

Lecture Notes in Computer Science 1838
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

*Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo*

Wieb Bosma (Ed.)

Algorithmic Number Theory

4th International Symposium, ANTS-IV
Leiden, The Netherlands, July 2-7, 2000
Proceedings

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Wieb Bosma
University of Nijmegen, Mathematical Institute
Postbus 9010, 6500 GL Nijmegen, The Netherlands
E-mail: bosma@sci.kun.nl

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Algorithmic number theory : 4th international symposium ; proceedings /
ANTS-IV, Leiden, The Netherlands, July 2 - 7, 2000. Wieb Bosma
(ed.) - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ;
London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2000
(Lecture notes in computer science ; Vol. 1838)
ISBN 3-540-67695-3

CR Subject Classification (1998): I.1, F.2.2, G.2, E.3-4, J.2

ISSN 0302-9743

ISBN 3-540-67695-3 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer is a company in the BertelsmannSpringer publishing group.
© Springer-Verlag Berlin Heidelberg 2000
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Christian Grosche, Hamburg
Printed on acid-free paper SPIN: 10722028 06/3142 5 4 3 2 1 0

Preface

The fourth Algorithmic Number Theory Symposium takes place at the Universiteit Leiden, in the Netherlands, from July 2-7, 2000. Its organization is a joint effort of Dutch number theorists from Leiden, Groningen, Nijmegen, and Amsterdam.

Six invited talks and 36 contributed talks are scheduled. This volume contains the written versions of the talks, with the exception of two of the invited talks. Not included are: *A rational approach to π* by Frits Beukers (Utrecht) and *The 40 trillionth binary digit of π is 0* by Peter Borwein (Burnaby, Canada). These talks are aimed at a wider audience, and form part of the special ANTS IV event *Pi in de Pieterskerk* on July 5, 2000. This event includes an evening ceremony in which the tombstone of Ludolph van Ceulen is replaced. Van Ceulen, who was appointed to Leiden in 1600, calculated 35 decimals of π . His tombstone in the Pieterskerk, in which these decimals were engraved, disappeared in the 19th century.

ANTS in Leiden is the fourth in a series of symposia that started in 1994. Previous locations were Cornell University, Ithaca, New York (1994), Université de Bordeaux I in Bordeaux, France (1996), and Reed College, Portland, Oregon (1998). The diversity of the papers contained in this volume shows that the main theme of ANTS, algorithmic number theory, is taken in a broad sense. The number of submissions for the Leiden conference largely exceeded the physical limitations of our one-week schedule. We are therefore confident that we are only at the beginning of a continuing tradition.

May 2000

Peter Stevenhagen
ANTS IV Program Chair
Wieb Bosma
Proceedings Editor

Organization

Organizing Committee

Wieb Bosma (Katholieke Universiteit Nijmegen)
Herman te Riele (CWI, Amsterdam)
Bart de Smit (Universiteit Leiden)
Peter Stevenhagen (Universiteit Leiden)
Jaap Top (Rijksuniversiteit Groningen)

Advisory Board

Dan Boneh (Stanford University)
Joe P. Buhler (Reed College, Portland)
Arjen K. Lenstra (Citibank)
Hendrik W. Lenstra, Jr. (UC Berkeley, and Universiteit Leiden)
Andrew M. Odlyzko (AT&T)
Rob Tijdeman (Universiteit Leiden)

Sponsoring Institutions

The organizers of ANTS IV gratefully acknowledge financial support from the following organizations.

Beegefonds, CWI
Centrum voor Wiskunde en Informatica
Koninklijke Nederlandse Akademie van Wetenschappen
Lorentz Center
Mathematical Research Institute
Rekenkamer Ludolph van Ceulen
Thomas Stieltjes Institute for Mathematics
Universiteit Leiden

Table of Contents

Invited Talks

- The Complexity of Some Lattice Problems 1
Jin-Yi Cai

- Rational Points Near Curves and Small Nonzero $|x^3 - y^2|$ via Lattice Reduction 33
Noam D. Elkies

- Coverings of Curves of Genus 2 65
E. Victor Flynn

- Lattice Reduction in Cryptology: An Update 85
Phong Q. Nguyen and Jacques Stern

Contributed Papers

- Construction of Secure C_{ab} Curves Using Modular Curves 113
Seigo Arita

- Curves over Finite Fields with Many Rational Points Obtained by Ray Class Field Extensions 127
Roland Auer

- New Results on Lattice Basis Reduction in Practice 135
Werner Backes and Susanne Wetzel

- Baby-Step Giant-Step Algorithms for Non-uniform Distributions 153
Simon R. Blackburn and Edlyn Teske

- On Powers as Sums of Two Cubes 169
Nils Bruin

- Factoring Polynomials over p -Adic Fields 185
David G. Cantor and Daniel M. Gordon

- Strategies in Filtering in the Number Field Sieve 209
Stefania Cavallar

- Factoring Polynomials over Finite Fields and Stable Colorings of Tournaments 233
Qi Cheng and Ming-Deh A. Huang

- Computing Special Values of Partial Zeta Functions 247
Gautam Chinta, Paul E. Gunnells, and Robert Sczech

VIII Table of Contents

Construction of Tables of Quartic Number Fields	257
<i>Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier</i>	
Counting Discriminants of Number Fields of Degree up to Four	269
<i>Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier</i>	
On Reconstruction of Algebraic Numbers	285
<i>Claus Fieker and Carsten Friedrichs</i>	
Dissecting a Sieve to Cut Its Need for Space	297
<i>William F. Galway</i>	
Counting Points on Hyperelliptic Curves over Finite Fields	313
<i>Pierrick Gaudry and Robert Harley</i>	
Modular Forms for $\mathrm{GL}(3)$ and Galois Representations	333
<i>Bert van Geemen and Jaap Top</i>	
Modular Symbols and Hecke Operators	347
<i>Paul E. Gunnells</i>	
Fast Jacobian Group Arithmetic on C_{ab} Curves	359
<i>Ryuichi Harasawa and Joe Suzuki</i>	
Lifting Elliptic Curves and Solving the Elliptic Curve Discrete Logarithm Problem	377
<i>Ming-Deh A. Huang, Ka Lam Kueh, and Ki-Seng Tan</i>	
A One Round Protocol for Tripartite Diffie–Hellman	385
<i>Antoine Joux</i>	
On Exponential Sums and Group Generators for Elliptic Curves over Finite Fields	395
<i>David R. Kohel and Igor E. Shparlinski</i>	
Component Groups of Quotients of $J_0(N)$	405
<i>David R. Kohel and William A. Stein</i>	
Fast Computation of Relative Class Numbers of CM-Fields	413
<i>Stéphane Louboutin</i>	
On Probable Prime Testing and the Computation of Square Roots mod n ..	423
<i>Siguna Müller</i>	
Improving Group Law Algorithms for Jacobians of Hyperelliptic Curves ..	439
<i>Koh-ichi Nagao</i>	
Central Values of Artin L -Functions for Quaternion Fields	449
<i>Sami Omar</i>	

The Pseudoprimes up to 10^{13}	459
<i>Richard G.E. Pinch</i>	
Computing the Number of Goldbach Partitions up to $5 \cdot 10^8$	475
<i>Jörg Richstein</i>	
Numerical Verification of the Brumer–Stark Conjecture	491
<i>Xavier-François Roblot and Brett A. Tangedal</i>	
Explicit Models of Genus 2 Curves with Split CM	505
<i>Fernando Rodriguez-Villegas</i>	
Reduction in Purely Cubic Function Fields of Unit Rank One	515
<i>Renate Scheidler</i>	
Factorization in the Composition Algebras	533
<i>Derek A. Smith</i>	
A Fast Algorithm for Approximately Counting Smooth Numbers	539
<i>Jonathan P. Sorenson</i>	
Computing All Integer Solutions of a General Elliptic Equation	551
<i>Roel J. Stroeker and Nikolaos Tzanakis</i>	
A Note on Shanks’s Chains of Primes	563
<i>Edlyn Teske and Hugh C. Williams</i>	
Asymptotically Fast Discrete Logarithms in Quadratic Number Fields	581
<i>Ulrich Vollmer</i>	
Asymptotically Fast GCD Computation in $\mathbb{Z}[i]$	595
<i>André Weilert</i>	
Author Index	615

The Complexity of Some Lattice Problems

Jin-Yi Cai*

Department of Computer Science and Engineering
State University of New York, Buffalo, NY 14260, USA
`cai@cse.buffalo.edu`

Abstract. We survey some recent developments in the study of the complexity of certain lattice problems. We focus on the recent progress on complexity results of intractability. We will discuss Ajtai's worst-case/average-case connections for the shortest vector problem, similar results for the closest vector problem and short basis problem, NP-hardness and non-NP-hardness, transference theorems between primal and dual lattices, and application to secure cryptography.

1 Introduction

Mostly stimulated by the recent work of Miklós Ajtai, there has been renewed interest and activity in the study of lattice problems. Research in the algorithmic aspects of lattice problems has been active in the past, especially following Lovász's basis reduction algorithm in 1982. The recent wave of activity and interest can be traced in large part to two seminal papers written by Miklós Ajtai in 1996 and in 1997 respectively.

In his 1996 paper [1], Ajtai found a remarkable worst-case to average-case reduction for some versions of the shortest lattice vector problem (SVP), thereby establishing a worst-case to average-case connection for these lattice problems. Such a connection is not known to hold for any other problem in NP believed to be outside P. In his 1997 paper [2], building on previous work by Adleman, Ajtai further proved the NP-hardness of SVP, under randomized reduction. The NP-hardness of SVP has been a long standing open problem. Stimulated by these breakthroughs, many researchers have obtained new and interesting results for these and other lattice problems [3,10,14,15,16,17,18,19,20,26,32,33,34,35,36,55], [58,61]. Our purpose in this article is to survey some of this development.

In my view these lattice problems are intrinsically interesting. Moreover, the worst-case to average-case connection discovered by Ajtai also opens up possibilities regarding provably secure public-key cryptography based on only worst-case intractability assumptions. It is well known that the existence of secure public-key cryptosystems presupposes $P \neq NP$. However the converse is far from being proven true.¹ The intractability required by cryptography is

* Research supported in part by grants from NSF CCR-9634665 and a John Simon Guggenheim Fellowship.

¹ I do not want to say "the converse is false", since it is probably *true* for the reason that both $P \neq NP$ and there exist secure public-key cryptosystems. But it is believed

more concerned with average-case complexity rather than worst-case complexity. Even if we assume that some problem in NP is not solvable in P or BPP, this still leaves open the possibility that the problem might be rather easy on the average.

Consider the security of RSA and the intractability of factoring. First, we do not know if factoring is not solvable in P or BPP. We do not know if this is so assuming $P \neq NP$. We do not even know whether it is NP-hard. Second, even if we assume it is NP-hard or not solvable in P or BPP, we do not know it is as hard for the special case of factoring a product of two large primes $p \cdot q$. Third, even if factoring $p \cdot q$ is hard in the worst case, we do not know if it is hard on the average, under some reasonable distribution on such numbers. Fourth, we do not know if decrypting RSA without the private key is equivalent to finding $\varphi(pq) = (p - 1)(q - 1)$, (although given $n = p \cdot q$, finding $\varphi(pq)$ is equivalent to factoring). Thus although RSA is believed to be an excellent public-key cryptosystem, there is a large gap between the assumption that factoring is hard in the worst-case (say it is not in BPP) and a proof that the system is secure.

Building on Ajtai's worst-case to average-case connection, Ajtai and Dwork [3] proposed a public-key cryptosystem that is provably secure, assuming only the worst case intractability of a certain version of SVP, namely to find the shortest lattice vector in a lattice with n^c -unique shortest vector, for a sufficiently large c . This is the first time that such a provable security guarantee based on the worst-case complexity alone has been established. However, for the important topic of application to Cryptology, Nguyen and Stern have written an excellent survey appearing in these proceedings [59]. Therefore I will not discuss this topic in any detail here and refer to [59].

In Section 2 we collect some definitions. I will then discuss Ajtai's worst-case/average-case connection for the shortest vector problem, and the worst-case/average-case connection for related closest vector problem (Section 3), NP-hardness results (Section 4), evidence of non-NP-hardness via bounded round interactive proof systems (Section 5), and transference theorems relating primal and dual lattices (Section 6).

I am sure many important works have been neglected or not given its proper due. I apologize for any such omissions or mistakes.

2 Preliminaries

A lattice is a discrete additive subgroup in some \mathbb{R}^n . Discreteness means that every lattice point is an isolated point in the topology of \mathbb{R}^n . An alternative definition is that a lattice consists of all the integral linear combinations of a set of linearly independent vectors,

$$L = \left\{ \sum_i n_i b_i \mid n_i \in \mathbb{Z}, \text{ for all } i \right\},$$

that it is insufficient to assume only $P \neq NP$ in order to prove pseudorandom number generators exist.

where the vectors b_i 's are linearly independent over \mathbb{R} . Such a set of generating vectors are called a basis. The dimension of the linear span, or equivalently the number of b_i 's in a basis is the rank (or dimension) of the lattice, and is denoted by $\dim L$. We may without loss of generality assume that $\dim L = n$, for otherwise we can replace \mathbb{R}^n by its linear span. We denote L as $L(b_1, b_2, \dots, b_n)$.

The basis of a lattice is not unique. Any two bases are related to each other by an integral matrix of determinant ± 1 . Such a matrix is called a unimodular matrix. Clearly an integral matrix has an integral inverse iff it is unimodular, following Cramer's rule.

The parallelepiped

$$P(b_1, \dots, b_n) = \{\sum x_i b_i \mid 0 \leq x_i < 1\}$$

is called the fundamental domain of the lattice.

Since basis transformation is unimodular, the determinant $|\det(b_1, \dots, b_n)|$ which is the volume of the fundamental domain $P(b_1, \dots, b_n)$ is independent of the basis, and is denoted by $\det(L)$.

We use lsp to denote linear span over \mathbb{R} . Given a basis $\{b_1, b_2, \dots, b_n\}$ of L , let $\Pi_i = \text{lsp}\{b_1, \dots, b_i\}$ be the linear span of $\{b_1, \dots, b_i\}$, and $L_i = L(b_1, \dots, b_i)$ be the sublattice generated by $\{b_1, \dots, b_i\}$. We denote by Π_i^\perp the orthogonal complement of Π_i . The process of Gram-Schmidt orthogonalization obtains from a basis $\{b_1, b_2, \dots, b_n\}$ a set of orthogonal vectors $\{\hat{b}_1, \hat{b}_2, \dots, \hat{b}_n\}$, where \hat{b}_i is the orthogonal component of b_i perpendicular to Π_{i-1} :

$$\hat{b}_i = b_i - \sum_{j < i} \frac{\langle b_i, \hat{b}_j \rangle}{\langle \hat{b}_j, \hat{b}_j \rangle} \hat{b}_j, \quad 1 \leq i \leq n,$$

where $\langle \cdot, \cdot \rangle$ denotes inner product.

The fundamental domain as well as the orthogonal “brick” $P(\hat{b}_1, \dots, \hat{b}_n) = [0, \hat{b}_1] \times \dots \times [0, \hat{b}_n]$ form a tessellation of \mathbb{R}^n by translation. We can also tessellate \mathbb{R}^n by the centralized “brick” $B = [-\frac{\hat{b}_1}{2}, \frac{\hat{b}_1}{2}] \times \dots \times [-\frac{\hat{b}_{i-1}}{2}, \frac{\hat{b}_{i-1}}{2}]$:

$$\mathbb{R}^n = \bigcup_{\ell \in L} (\ell + B).$$

We note that the volume $\text{vol } D = \text{vol } B = \det L$.

The length of the shortest non-zero vector of L is denoted by $\lambda_1(L)$. In general, Minkowski's successive minima $\lambda_i(L)$ are defined as follows: for $1 \leq i \leq \dim L$,

$$\lambda_i(L) = \min_{v_1, \dots, v_i \in L} \max_{1 \leq j \leq i} \|v_j\|,$$

where the sequence of vectors $v_1, \dots, v_i \in L$ ranges over all i linearly independent lattice vectors. It is not difficult to show that to get $v_i \in L$ with $\|v_i\| = \lambda_i$, one can always take greedily *any* linearly independent $v_1, \dots, v_{i-1} \in L$, with $\|v_1\| = \lambda_1, \dots, \|v_{i-1}\| = \lambda_{i-1}$.

Let L be an n -dimensional lattice in \mathbb{R}^n with basis $\{b_1, b_2, \dots, b_n\}$. Since the translations of the fundamental domain $D = P(b_1, b_2, \dots, b_n)$ form a tiling of \mathbb{R}^n , the volume $\text{vol}(D) = \det(L)$ provides a certain measure of the size of L . Minkowski's First Theorem makes an explicit connection of the shortest lattice vector and this quantity [57,24,39]:

Theorem 1 (Minkowski).

$$\lambda_1(L) \leq \gamma_n (\det(L))^{1/n},$$

where γ_n is some universal constant.

The smallest such constant for dimension n is denoted by γ_n and called Hermite's constant of rank n . Minkowski proved that $\gamma_n \leq \frac{2}{\sqrt{\pi}} \Gamma(\frac{n}{2} + 1)^{1/n}$, which is asymptotically $\sqrt{\frac{2n}{\pi e}}$. It is known that $\sqrt{\frac{n}{2\pi e}} \leq \gamma_n \leq \sqrt{\frac{n}{\pi e}}$. The upshot is, for a lattice with $\det(L) = 1$, (after a suitable scaling), there is always a non-zero short vector of length no more than \sqrt{n} .

Minkowski's First Theorem has a short and elegant proof: Consider the lattice $L' = 2L$, which is a dilatation of L by a factor of 2 in all directions. $\det(L') = 2^n \det(L)$. Consider a ball of radius r centered at every lattice point of L' . Let ω_n denote the volume of a unit ball B_n , then $\omega_n r^n$ is the volume of a ball $B_n(r)$ of radius r . Now if $\omega_n r^n > \det(L')$, there must be some overlap among two different balls, thus $\exists \ell \neq \ell' \text{ both } \in L$, such that $2\ell + x = 2\ell' + y$ for some $x, y \in B_n(r)$. Then $\ell - \ell' = (y - x)/2 \in B_n(r)$ by convexity. And $\ell - \ell'$ is our non-zero lattice point of L . It is known that $\omega_n = \pi^{n/2}/\Gamma(\frac{n}{2} + 1)$. It follows that

$$\lambda_1(L) \leq \frac{2}{\sqrt{\pi}} \Gamma(\frac{n}{2} + 1)^{1/n} (\det(L))^{1/n} = \Theta(\sqrt{n}) (\det(L))^{1/n}.$$

Theorem 1 follows.

A more general theorem, also due to Minkowski, is concerned with successive minima:

Theorem 2 (Minkowski).

$$\left(\prod_{i=1}^n \lambda_i(L) \right)^{1/n} \leq \Theta(\sqrt{n}) (\det(L))^{1/n}.$$

While Minkowski's theorem guarantees the existence of vectors as short as $\sqrt{n} \det(L)^{1/n}$, there is no polynomial-time algorithm to find such a vector. Minkowski's proof is decidedly non-constructive. The Shortest Vector Problem (SVP) is the following: Given a basis of L , find a vector $v \in L$ such that $\|v\| = \lambda_1(L)$. One can also define various approximate short vector problems, seeking a non-zero $v \in L$ with $\|v\|$ bounded by some approximation factor, $\|v\| \leq f(n)\lambda_1(L)$ or $\|v\| \leq f(n)(\det(L))^{1/n}$.

We denote by $\text{bl}(L)$ the basis length of L

$$\text{bl}(L) = \min_{\text{all bases } b_1, \dots, b_n \text{ for } L} \max_{i=1}^n \|b_i\|.$$

The dual lattice L^* of a lattice L of dimension n in \mathbb{R}^n is defined as those vectors $u \in \mathbb{R}^n$, such that $\langle u, v \rangle \in \mathbb{Z}$, for all $v \in L$. For a basis $\{b_1, b_2, \dots, b_n\}$ of L , its dual basis is $\{b_1^*, b_2^*, \dots, b_n^*\}$, where $\langle b_i^*, b_j \rangle = \delta_{ij}$. Then $L^* = L(b_1^*, b_2^*, \dots, b_n^*)$. In particular $\det(L^*) = 1/\det(L)$, and $L^{**} = L$. For a lattice with dimension less than n , its dual is defined within its own linear span.

We let $kL = \{kv \mid v \in L\}$ be the dilatation of L for any positive $k \in \mathbb{R}$. Let $x + A = \{x + y \mid y \in A\}$ for any $x \in \mathbb{R}^n$ and $A \subseteq \mathbb{R}^n$. Let $A + B = \{a + b \mid a \in A, b \in B\}$. We denote by $\lfloor x \rfloor$ the greatest integer $\leq x$, $\lceil x \rceil$ the least integer $\geq x$, $\lceil x \rceil = -\lfloor -x \rfloor$, and $\lceil x \rceil$ the closest integer to x , $\lceil x \rceil = \lfloor x + \frac{1}{2} \rfloor$.

3 Ajtai's Worst-Case to Average-Case Connection

Let n, m and q be arbitrary integers. Let $\mathbb{Z}_q^{n \times m}$ denote the set of $n \times m$ matrices over \mathbb{Z}_q , and let $\Omega_{n,m,q}$ denote the uniform distribution on $\mathbb{Z}_q^{n \times m}$. For any $X \in \mathbb{Z}_q^{n \times m}$, the set $\Lambda(X) = \{y \in \mathbb{Z}^m \mid Xy \equiv 0 \pmod{q}\}$ (where the congruence is component-wise) defines a lattice of dimension m . Let $\Lambda = \Lambda_{n,m,q}$ denote the probability space of lattices consisting of $\Lambda(X)$ by choosing X according to $\Omega_{n,m,q}$.

We note that indeed $\Lambda(X)$ is a lattice of dimension m , since it is clearly a discrete additive subgroup of \mathbb{Z}^m , and each $qe_i \in \Lambda(X)$, where e_i has a single 1 at the i th position and 0 elsewhere. It also follows that $\Lambda(X)$ repeats itself within each $q \times q \times \dots \times q$ box. In other words, $\Lambda(X)$ is invariant under the translations $y \mapsto y + qe_i$, for each $1 \leq i \leq m$.

By Minkowski's First Theorem, it can be shown that

$$\forall c \exists c' \text{ s.t. } \forall \Lambda(X) \in \Lambda_{n,c'n,n^c} \exists v (v \in \Lambda(X) \text{ and } 0 < \|v\| \leq n).$$

In fact the bound n can be reduced to $n^{\frac{1}{2}+\epsilon}$. The bound $\|v\| \leq n$ is needed to ensure that the assumption on the hypothetical algorithm \mathcal{A} below is non-vacuous.

Theorem 3 (Ajtai). *Suppose there is a probabilistic polynomial time algorithm \mathcal{A} such that for all n , when given a random lattice $\Lambda(X) \in \Lambda_{n,m,q}$ where $m = \alpha n \log n$ and $q = n^\beta$ for appropriate constants α, β , returns with probability $\frac{1}{n^{O(1)}}$, a vector of $\Lambda(X)$ of length $\leq n$, then there exists a probabilistic polynomial time algorithm \mathcal{B} such that for all n , when given a basis $\{a_1, \dots, a_n\}$ for an arbitrary lattice $L = L(a_1, \dots, a_n)$, performs the following with high probability:*

- 1) Finds a basis $\{b_1, \dots, b_n\}$ for L such that

$$\max_{i=1}^n \|b_i\| \leq n^{c_1} \cdot \text{bl}(L),$$

- 2) Finds an estimate $\tilde{\lambda}$ of $\lambda_1(L)$ such that,

$$\frac{\lambda_1(L)}{n^{c_2}} \leq \tilde{\lambda} \leq \lambda_1(L),$$

- 3) Finds the unique shortest vector $\pm v$ of L , if L has an n^{c_3} unique shortest vector, i.e. $\lambda_2(L) \geq n^{c_3} \cdot \lambda_1(L)$,

where c_1, c_2, c_3 are absolute constants.

Remark: This is the first such worst-case to average-case connection proved for a problem in NP believed not in P. While random-self-reducibilities were known for other problems, such as Quadratic Residuosity (QR), there is a technical difference. In QR, one must fix a modulus, then there is a worst-case to average-case connection for this modulus. But no such reduction is known among different moduli. The permanent is another example where there is a certain worst-case to average-case connection (see [31,30,21,38]), but the permanent is not believed to be in NP.

Items 2) and 3) are derived from item 1) via a transference type argument, about which we will say more later in Section 6. Here we will focus on the ideas in the proof of item 1). Without loss of generality, we can assume that the lattice consists of integral vectors. The same result also holds for lattices with rational entries or with entries from any subfield of \mathbb{C} , as long as there is an effective bit representation for the lattice.

We will now present some ideas from the proof.

Suppose we currently have a basis $\{b_1, \dots, b_n\}$, where $\max_{i=1}^n \|b_i\|$ is greater than $\text{bl}(L)$ by a large polynomial factor n^{c_1} , i.e.

$$\mu \equiv \max_{i=1}^n \|b_i\| > n^{c_1} \text{bl}(L).$$

The main procedure of \mathcal{B} is iterative. Let S be a set of n independent vectors of L (initially $S = \{b_1, \dots, b_n\}$). If the length of the elements of S at the start of the current iteration is large enough, the algorithm finds a set of independent vectors, each of at most half the length, with high probability. This means, in a polynomial number of steps we will have a set of short enough vectors, which can then be converted to a short basis with a loss of a factor $\leq \sqrt{n}$.

The fundamental domain $D = P(b_1, \dots, b_n)$ forms a tiling of \mathbb{R}^n via translations under L ,

$$\mathbb{R}^n = \bigcup_{l \in L} (l + D),$$

as a disjoint union.

Consider a large cube

$$Q = \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^n x_i e_i, 0 \leq x_i < M\},$$

where M is a certain polynomial factor greater than μ , say, $M = n^\gamma \mu$. For each i , we can “round” the corner point Me_i to a lattice point according to which translate $l_i + D$ it belongs to. This only involves solving a linear system expressing Me_i as a rational linear combination of the basis $\{b_1, \dots, b_n\}$ and

then rounding the coordinates. Thus for each $i = 1, \dots, n$, let

$$Me_i = \sum_{j=1}^n \alpha_{ij} b_j \quad \text{and} \quad l_i = \sum_{j=1}^n \lfloor \alpha_{ij} \rfloor b_j.$$

Now

$$Q' = \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^n x_i l_i, 0 \leq x_i < 1\},$$

is a reasonably good approximation of Q ; we will call it a pseudocube. Note that the corner vertices of Q' are all lattice points. To ensure that Q' looks reasonably close to a cube, Ajtai chose $\gamma = 3$.

In the next step we subdivide Q' into a family of disjoint sub-pseudocubes, by subdividing Q' along each direction l_i into q subintervals, where q is polynomially bounded in n .

$$Q' = \bigcup_{0 \leq k_1, \dots, k_n < q} \left(\sum_{i=1}^n \frac{k_i}{q} l_i + Q'' \right),$$

where the basic sub-pseudocube

$$Q'' = \{x \in \mathbb{R}^n \mid x = \sum_{i=1}^n x_i l_i, 0 \leq x_i < \frac{1}{q}\}.$$

We will make sure that the length of a side of Q'' , which is roughly $\frac{M}{q}$, is still larger than $\text{bl}(L)$ by a significant polynomial factor.

Suppose this is the case. Then with a series of technical lemmas, Ajtai shows that the number of lattice points within each translate $Q'' + \sum_{i=1}^n \frac{k_i}{q} l_i$ is roughly the same. This is intuitively quite plausible. But the technical details are not straightforward, especially if one wants a reasonably good bound. (See below.)

Once this approximate equi-distribution of lattice points is established, one can sample the “addresses” (k_1, \dots, k_n) of sub-pseudocubes, by uniformly sampling a lattice point in Q' . Once a lattice point v is picked, we decide to which sub-pseudocube it belongs by expressing v as a linear combination $\sum_{i=1}^n \frac{x_i}{q} l_i$, where $0 \leq x_i < q$, by solving a linear system. Then, we round off x_i and set $k_i = \lfloor x_i \rfloor$.

More generally, suppose we get m such samples, $v_j \in L$, $1 \leq j \leq m$. We decompose v_j as follows, (See Figure 1)

$$v_j = \sum_{i=1}^n \frac{k_{ij}}{q} l_i + r_j,$$

where r_j is a vector in Q'' . Note that $\|r_j\|_2$ is $O(\frac{\sqrt{n}M}{q})$.

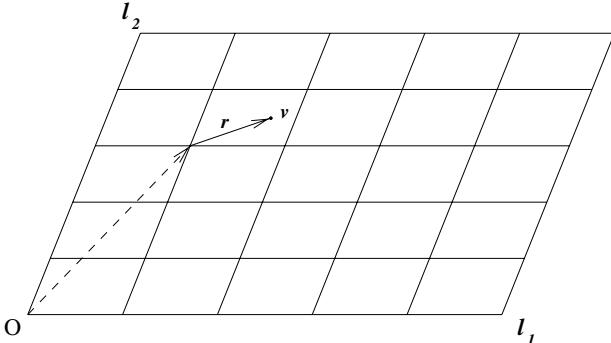


Figure 1

Here is the key observation: Suppose we are able to obtain an integral solution $X = (\xi_1, \dots, \xi_m)$ to

$$\sum_{j=1}^m k_{ij}\xi_j \equiv 0 \pmod{q},$$

then $\sum_{j=1}^m \xi_j v_j$ would be a lattice point which has an interesting decomposition,

$$\sum_{j=1}^m \xi_j v_j = \sum_{i=1}^n \left(\frac{\sum_{j=1}^m k_{ij}\xi_j}{q} \right) l_i + \sum_{j=1}^m \xi_j r_j. \quad (1)$$

We note that the quantity $\frac{\sum_{j=1}^m k_{ij}\xi_j}{q}$ is actually an integer, which makes the first term in (1) a lattice vector. Hence $\sum_{j=1}^m \xi_j r_j$, being the difference of two lattice points, must be a lattice point itself, (even though each r_j is probably not a lattice point.)

Suppose the integral solution X has every $|\xi_j| \leq n$, then

$$\begin{aligned} \left\| \sum_{j=1}^m \xi_j r_j \right\| &\leq m \cdot n \cdot O\left(\frac{\sqrt{n}M}{q}\right) \\ &= O\left(\frac{m \cdot n^{1.5+\gamma}}{q} \mu\right). \end{aligned} \quad (2)$$

Now q can be chosen $\Theta(n^6)$ so that $\left\| \sum_{j=1}^m \xi_j r_j \right\| < \frac{\mu}{2}$, which is at most half of every $\|b_i\|$.

With the choice of $\gamma = 3$, Ajtai showed that the shape of the pseudocube and thus that of the sub-pseudocubes is very close to a perfect cube. With a choice of $q = \Theta(n^6)$, and a corresponding $m = O(n \log n)$, Minkowski's Theorem applies. Hence the assumption on \mathcal{A} is non-vacuous and the newly produced

lattice vector $\sum_{j=1}^m \xi_j r_j$ has length $< \frac{\mu}{2}$. On the other hand, the length of a side of a sub-pseudocube is approximately $\frac{M}{q}$ which is bounded below by $\frac{n^{\gamma+c_1}}{q} \text{bl}(L) = \Theta(n^{c_1-3} \text{bl}(L))$.

With the shape of the pseudocube approximately a perfect cube, and with a sufficiently large c_1 , which makes each side of the sub-pseudocube sufficiently larger than $\text{bl}(L)$, Ajtai showed that the distribution induced on the address space $\{(k_1, \dots, k_n) \mid 0 \leq k_i < q\}$ by uniformly sampling lattice points from L is close to uniform. In fact, not only must the distribution of each sample (k_1, \dots, k_n) be close to uniform, but also the joint distribution on all the m samples forming the matrix (k_{ij}) must be close to the uniform distribution $\Omega_{n,m,q}$. Only then can one legitimately invoke the assumed algorithm \mathcal{A} and be guaranteed to obtain a short vector X with $\sum_{j=1}^m k_{ij} \xi_j \equiv 0 \pmod{q}$, and $\|X\| \leq n$, with nontrivial probability.

So far we have only produced one lattice vector $b'_1 = \sum_{j=1}^m \xi_j r_j$, which is shorter than $\mu = \max \|b_i\|$ by a factor of 2. We continue this process to produce n linearly independent lattice vectors $\{b'_1, \dots, b'_n\}$ to replace $\{b_1, \dots, b_n\}$. To show that these successive b'_i are linearly independent demands another set of technical lemmas which ultimately depend on the fact that c_1 is sufficiently large. In that case, Ajtai showed that within each sub-pseudocube the lattice is quite dense. It follows that, for every $n - 1$ dimensional hyperplane Π , the number of lattice points on $\Pi \cap Q''$ is much smaller compared to the total number of lattice points in Q'' . Moreover, this is true for every translate of Q'' . It follows that the successive b'_i 's are not likely to be linearly dependent on $\{b'_1, \dots, b'_{i-1}\}$. We will not provide any more technical details of Ajtai's proof. The interested reader is referred to [1].

Improving Ajtai's Connection Factors

What is outlined above is essentially Ajtai's proof [1], where some universal constants c_1, c_2 and c_3 are shown. Although no explicit values for these c_i 's were given, and apparently no special effort was made to minimize them, implicitly a factor less than 8, 10 and 19, respectively, can be derived from the proofs of [1].

The factors n^{c_i} are called Ajtai's connection factors; they provide a measure of the tightness of the worst-case to average-case connection. The smaller the constants are, the tighter the connection one gets. As 2) and 3) are derived through 1) (see Section 6), n^{c_1} is the crucial factor. Cai and Nerurkar [19] obtained a substantial improvement to n^{c_1} , and consequently to the other factors as well. Here we give an overview of some of the ideas involved in this improvement. As is the case with Ajtai's proof [1], there are a number of technical points we have to gloss over due to limited space.

The general structure of the procedure of Cai and Nerurkar [19] closely follows Ajtai's proof, but much of the technical justification is different. As we saw above, the general idea is to sample lattice points, in order to induce an almost uniform distribution on a set of "address" vectors, which form the columns of a matrix that is close to uniformly distributed. The assumed algorithm \mathcal{A} is applied to

this matrix. By hypothesis, this algorithm performs well on the average, and thus we get a short vector which can be turned into a short vector of the original lattice.

In the choice of $M = n^\gamma \mu$, we need γ to be a sufficiently large constant in order to ensure that the resulting pseudocube is reasonably close to a perfect cube. We call this the shape condition. Then, we need to choose an integer q to be a sufficiently large polynomial (in n) in order to ensure that the newly produced remainder vector is shorter than the previous $\|b_i\|$. This involves m in the numerator in $n^{\gamma+1.5}m/q$ in (2), which has to be chosen after q in order to ensure that short vectors exist by Minkowski's First Theorem. Fortunately, this is not circular; for any polynomially bounded q , m only needs to be $O(n)$. But still q must depend on γ . Finally, given q , we must ensure that the length of a side of a sub-pseudocube M/q is sufficiently large compared to $\text{bl}(L)$. We know that,

$$\frac{M}{q} = \frac{n^\gamma \mu}{q} > \frac{n^{\gamma+c_1}}{q} \text{bl}(L)$$

This is where $\mu > n^{c_1} \text{bl}(L)$ is used and c_1 has to be large. Cai and Nerurkar [19] achieve $c_1 = 3 + \epsilon$ for linearly independent vectors, and $c_1 = 3.5 + \epsilon$ for basis length.

The algorithmic improvement by Cai and Nerurkar [19] starts with a tiling of \mathbb{R}^n by orthogonal ‘‘bricks’’ of sides at most μ , via Gram-Schmidt orthogonalization. This is in contrast to the tiling by fundamental domains in [1]. The advantage is that one can round off from a perfect cube to a lattice pseudocube with less error. Thus, for $M = n^{1.5}\mu$ and $w_i = Me_i$, we can round off w_i to a lattice point l_i such that $w_i = l_i + \delta_i$ and $\|\delta_i\| \leq \frac{\sqrt{n}\mu}{2}$. This implies $\|l_i\| \leq (n^{1.5} + \frac{\sqrt{n}}{2})\mu$. $P(l_1, \dots, l_n)$ is the pseudocube constructed.

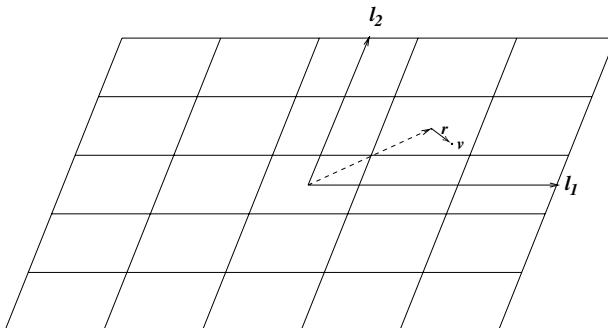


Figure 2

Secondly, in [19], the pseudocube is positioned centrally and subdivided. Each sub-pseudocube will have an address vector at the center. More precisely we will take $Q' = P(2l_1, \dots, 2l_n) - \sum_{i=1}^n l_i = \{\sum_{i=1}^n z_i l_i \mid -1 \leq z_i < 1\}$. We partition Q' into q^n sub-pseudo-cubes, (where q is odd, say), such that the basic sub-pseudocube is $Q' = \{\sum_{i=1}^n z_i l_i \mid -\frac{1}{q} \leq z_i < \frac{1}{q}\}$. We will sample lattice points uniformly in the pseudocube Q' . This induces an almost uniform distribution

on the address space. But this time we consider each address as corresponding to the *center* of the sub-pseudocube. When we express a sample lattice point v_j as the sum of this address vector and a remainder vector r_j , these remainder vectors tend to be symmetrically distributed with respect to the address vector at the center. (See Figure 2)

Here an address vector is of the form $\sum_{i=1}^n \frac{k_{ij}}{q} l_i$, where each k_{ij} is even, $-(q-1) \leq k_{ij} \leq q-1$. The corresponding “address” is $(k_{1j}, k_{2j}, \dots, k_{nj})$ reduced modulo q . Thus, when we estimate $\|\sum_{j=1}^m \xi_j r_j\|$ probabilistically, the independent r_j 's tend to cancel out instead of adding up. Note that $X = (\xi_1, \dots, \xi_m)$ is a (short) solution obtained by the algorithm \mathcal{A} given only the address matrix (k_{ij}) . Given such a matrix one must ensure that the r_j are almost independently and centrally symmetrically distributed. This is geometrically quite intuitive, given a sufficiently large ratio of the sides of the sub-pseudocube to $\text{bl}(L)$. But the hard part is to minimize this notion of “sufficiently large”. It turns out that $q = n^{3+\epsilon}$ and $\mu > n^{3+\epsilon}\text{bl}(L)$ will do. The technical part of the proof is rather involved.

There is one more idea in [19] in the improvements in terms of the algorithmic steps. It turns out to be insufficient to guarantee the generation of one almost uniform address vector, which makes up one column of the matrix. We must be able to generate m columns to form an almost uniformly generated matrix. This more stringent requirement is needed to apply the algorithm \mathcal{A} . In [19] we used an idea to amplify the “randomness” in each column vector generated, by adding together $\lceil 2/\epsilon \rceil$ copies of independent samples

$$\begin{aligned} v &= \sum_{i=1}^n \frac{k_i}{q} l_i + r, \\ v' &= \sum_{i=1}^n \frac{k'_i}{q} l_i + r', \quad \text{etc.} \end{aligned}$$

This gives a lattice point

$$v + v' + \dots = \sum_{i=1}^n \frac{k_i + k'_i + \dots}{q} l_i + (r + r' + \dots).$$

Starting from the column vector (k_1, k_2, \dots, k_n) being $n^{-\epsilon}$ -close to uniform, we show that the address vector

$$(k_1 + k'_1 + \dots, k_2 + k'_2 + \dots, \dots, k_n + k'_n + \dots) \bmod q$$

is n^{-2} -close to uniform, which would be sufficient to ensure that the matrix is close to being uniform. The price we pay for this is that each remainder vector is enlarged by a factor at most $\lceil 2/\epsilon \rceil$.

The more difficult part of the proof is to show that the lattice samples do induce a distribution that is $n^{-\epsilon}$ -close to uniform on the address space. In addition to our “shape condition”, which is accomplished by $\gamma = 1.5$, we need to

estimate the volume of each sub-pseudocube to ensure that the number of lattice points within each sub-pseudocube is almost identical. Moreover, in order to obtain independent lattice vectors, we need to ensure that the proportion of lattice points in a sub-pseudocube that lie on any (co-1 dimensional) hyperplane is negligible.

The bounds in [19] use eigenvalues and singular values, and a theorem of K. Ball [7]. We cannot go into much detail here, but the following lemmas give a flavor of it.

Lemma 1. *Let e_1, \dots, e_n be the standard unit vectors. Let u_1, \dots, u_n be linearly independent vectors such that $\|u_i - e_i\| \leq \epsilon$. Then the parallelepiped $P(u_1, \dots, u_n)$ has volume*

$$1 - n\epsilon \leq \text{vol}(\mathcal{P}(u_1, \dots, u_n)) \leq (1 + \epsilon)^n.$$

(One cannot improve the lower bound to $(1 - \epsilon)^n$ for large n .)

Lemma 2. *Let e_1, \dots, e_n and u_1, \dots, u_n be as above. Let H be a hyperplane. Then the $(n - 1)$ -dimensional volume of $P(u_1, \dots, u_n) \cap H$ is at most $\sqrt{2e}(1 + \epsilon)^{n-1}$.*

A Worst-Case/Average-Case Connection for CVP

The best bound for the hardness of CVP is by Dinur et. al. [26]. They show that CVP is NP-hard to approximate within a factor $2^{\log^{1-\epsilon} n}$, for an $\epsilon = o(1)$. Goldreich et. al. [36] show a direct reduction from SVP to CVP. This reduction has the property of preserving the factor of approximation for the two problems and the dimension of the lattice.

A corresponding result of worst-case/average-case connection for CVP was established by Cai [23] recently. Note that the known NP-hardness reductions to CVP do not provide any evidence of hardness for the average-case complexity of CVP. This is generally true for NP-hardness reductions, since the reductions only produce very specialized instances of the target problem, in this case CVP.

Theorem 4. *If there is a probabilistic P-time algorithm \mathcal{A} , for a uniformly chosen lattice L in the class Λ indexed by n and a uniformly chosen target vector u , \mathcal{A} finds a lattice vector $v \in L$ such that $\|u - v\| < n$, with probability at least $1/n^{O(1)}$, then, there is a probabilistic P-time algorithm \mathcal{B} , for any lattice L' of dimension N , with probability $1 - e^{-N}$, will*

- For any target vector x find a lattice vector $y \in L'$ with distance $\|x - y\| < N^{c_1} \text{bl}(L')$;
- Find an estimate $\tilde{\lambda}$ of the shortest lattice vector length $\lambda_1(L')$ such that,

$$\frac{\lambda_1(L')}{N^{c_2}} \leq \tilde{\lambda} \leq \lambda_1(L');$$

- Find the unique shortest vector $\pm v$ of L' , if L' has an N^{c_3} -unique shortest vector; and

- Find a basis b_1, b_2, \dots, b_N , such that the maximum length $\max_{1 \leq i \leq N} \|b_i\| < N^{c \text{bl}}(L')$;

where c_1, c_2, c_3 and c are absolute constants.

4 NP-Hardness

It was known that SVP, under the l_∞ -norm, is NP-hard [47,66]. It was also shown there that the related Closest Vector Problem (CVP) is NP-hard for all l_p -norms, $p \geq 1$. Arora et al. [5] showed that, under any l_p -norm, CVP is NP-hard to approximate within any constant factor, and that if it can be approximated within a factor of $2^{\log^{1/2-\epsilon} n}$, then NP is in quasi-polynomial time.

It had long been thought that the Shortest Vector Problem for the natural l_2 -norm is NP-hard. This was conjectured e.g., by Lovász [52]. It remained a major open problem until, in 1997, Ajtai [2] proved the NP-hardness of the SVP for this norm, under randomized reductions. Moreover, Ajtai showed that to approximate the shortest vector of an n -dimensional lattice within a factor of $(1 + \frac{1}{2^{n^k}})$ (for a sufficiently large constant k) is also NP-hard under randomized reductions. This was improved to $(1 + \frac{1}{n^\epsilon})$ for any constant $\epsilon > 0$ by Cai and Nerurkar [20], and then to any constant smaller than $\sqrt{2}$ by Micciancio [55].

Theorem 5. *It is NP-hard, under randomized polynomial time reductions, to find a shortest lattice vector, even to approximate it within a factor of $\sqrt{2} - \epsilon$, for any $\epsilon > 0$.*

In the next subsection we outline Ajtai’s result. The presentation incorporates the simplifications and improvements of [20] but the main ideas are due to Ajtai. After that we present Micciancio’s improvement.

Ajtai’s Result

Ajtai gave a randomized reduction from the following variant of the subset sum problem to SVP.

The Restricted Subset Sum Problem. Given integers a_1, \dots, a_l, A , each of bit-length $\leq l^3$, find a 0-1 solution to the system $\sum_{i=1}^l a_i x_i = A$ and $\sum_{i=1}^l x_i = \lfloor \frac{l}{2} \rfloor$.

We first define a lattice which will play a crucial role in the proof. This lattice is a modified version of the one used by Adleman (unpublished) in his reduction from factoring to the SVP, under some unproven assumptions. For this lattice, we need to choose several parameters depending on the l in the restricted subset sum instance.

- n is chosen to be a sufficiently large polynomial in l .
- m is chosen to be a sufficiently large polynomial in n .
- $m \gg n \gg l$.

- b is chosen randomly from the set of products of n distinct elements of $\{p_1, \dots, p_m\}$, the first m primes.
- ω is chosen a constant root of b .
- B is polynomial in ω .

Clearly, B, b and ω are exponential in n . We will not be overly precise here about the values of these parameters in order not to obscure the main points. Using these parameters, Ajtai defines the following matrix, whose $m+2$ columns generate a lattice.

$$\begin{pmatrix} \sqrt{\log p_1} & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & \sqrt{\log p_m} & 0 & 0 \\ 0 & \cdots & 0 & 0 & \omega^{-2} \\ B \log p_1 & \cdots & B \log p_m & B \log b & B \log \left(1 + \frac{\omega}{b}\right) \end{pmatrix}$$

Lattice L_A

This lattice is then normalized. The normalized lattice has every vector of length at least 1 and a lot of vectors of length very close to 1. We will denote by v_i , the columns of the basis matrix for this modified lattice. We will denote this normalized matrix, as well as the lattice it generates, by L . With high probability, this lattice, $L = L(v_1, \dots, v_{m+2})$, has the interesting properties we outline next. These properties are a consequence of the way primes are distributed and the convexity of the logarithm function.

1. All non-zero vectors have length at least 1.
2. There are a lot of vectors of small norm with the property that their first m basis coefficients $\in \{0, -1\}$. More precisely, let Y be the set of all $v \in L$, $v = \sum_{i=1}^{m+2} \alpha_i v_i$ with $\sum_{i=1}^m |\alpha_i| = n$, $\alpha_i \in \{0, -1\}$ for $i \in \{1, \dots, m\}$, and $\|v\|^2 < 1 + \delta$. Then $|Y| \geq 2^{n \log n}$. Here, δ is an exponentially small quantity.
3. Any two distinct elements of Y differ in their first m basis coefficients.
4. If v is a non-zero vector of L of squared norm less than $1 + \frac{2}{m^{3\epsilon/4}}$, then the first $m+1$ coefficients of v have a special form. More precisely, if $v = \sum_{i=1}^{m+2} \alpha_i v_i$, $\|v\|^2 < 1 + \frac{2}{m^{3\epsilon/4}}$, and $\alpha_{m+1} \geq 0$, then $\alpha_1, \dots, \alpha_m \in \{0, -1\}$ and $\alpha_{m+1} = 1$.

This lattice is now extended in the following random manner depending on the given instance of the restricted subset sum problem. With high probability, given a reasonably short vector in this extended lattice, a solution to the instance can be produced.

Let $\sum_{i=1}^l a_i x_i = A$ be the given instance of the restricted subset sum problem. Let $\epsilon > 0$ be any constant. Let $\tau = 2/m^\epsilon$ and $\beta = \sqrt{\tau}$. Let $C = C_1, \dots, C_l$ be a random sequence of pairwise disjoint subsets of $\{1, \dots, m\}$. Define an $(l+2) \times (m+2)$ matrix D as follows. The $(m+2)^{\text{nd}}$ column is all zeros. The $(m+1)^{\text{st}}$ column is $(Al\beta, \lfloor \frac{l}{2} \rfloor l\beta, 0, \dots, 0)^T$. The other entries of the matrix are defined in the following manner.

1. The first row has the entry $a_i l\beta$ in the j^{th} position if $j \in C_i$, and otherwise has zero.

2. The second row has the entry $l\beta$ in the j^{th} position if j is in some C_i , and otherwise has zero.
3. For i from 3 to $l+2$, row i has β in the j^{th} position if $j \in C_{i-2}$ and otherwise has zero.

If C_1, \dots, C_l are consecutive intervals of $\{1, \dots, m\}$, then D is the following matrix,

$$\begin{pmatrix} a_1 l \beta & \cdots & a_1 l \beta & \cdots & a_1 l \beta & \cdots & A l \beta & 0 \\ l \beta & \cdots & l \beta & \cdots & l \beta & \cdots & \lfloor \frac{l}{2} \rfloor l \beta & 0 \\ \beta & \cdots & \beta & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & & \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & \beta & \cdots & \beta & 0 \\ & & & & & & & 0 \end{pmatrix}$$

The extended lattice is the lattice $L^{(D)}$ generated by the columns of the matrix $\begin{pmatrix} L \\ D \end{pmatrix}$. A vector $\bar{v} \in L^{(D)}$ can be written $\begin{bmatrix} v \\ v' \end{bmatrix}$, where for some integral column vector $\alpha = (\alpha_1, \dots, \alpha_{m+2})^T$, $v = \sum_{i=1}^{m+2} \alpha_i v_i \in L$ and $v' = D\alpha$. Each v uniquely determines its α and thus uniquely determines v' .

Ajtai uses a constructive variant of the following combinatorial lemma, due to Sauer, to show that any solution to a subset sum instance can be produced from the coefficients of some short vector. A proof of this lemma can be found, for example, in [4].

Lemma 3 (Sauer). *Let S be a finite set and \mathcal{S} be a set of subsets of S . If for some k , $|\mathcal{S}| > \sum_{i=1}^k \binom{|S|}{i}$, then there is a $X \subseteq S$ with k elements such that $2^X = \{X \cap Z \mid Z \in \mathcal{S}\}$.*

That is, every subset of X can be realized by intersecting it with some element of \mathcal{S} . A consequence of Ajtai's constructive lemma is that a random sequence $C = C_1, \dots, C_l$ of subsets of $\{1, \dots, m\}$, has the following property:

$$\forall s \in \{0, 1\}^l, \exists v = \sum_{j=1}^{m+2} \alpha_j v_j \in Y \text{ such that, } \forall i \in \{1, \dots, l\}, s_i = -\sum_{j \in C_i} \alpha_j.$$

This property implies that if there is a solution to the restricted subset sum instance then there is a vector in the set Y that gives rise to it. That is, suppose $\sum_{i=1}^l a_i x_i = A$ has a solution $x_i = s_i$, i.e.

$$s_i \in \{0, 1\}, \quad \sum_{i=1}^l a_i s_i = A \quad \text{and} \quad \sum_{i=1}^l s_i = \lfloor \frac{l}{2} \rfloor.$$

Then, $\exists v \in Y, v = \sum_{j=1}^{m+2} \alpha_j v_j$, such that $\forall i \in \{1, \dots, l\}$,

$$s_i = -\sum_{j \in C_i} \alpha_j.$$

Since $v \in Y$, $0 < \|v\|^2 \leq 1 + \delta$. Let $\bar{v} \in L^{(D)}$, $\bar{v} = \begin{bmatrix} v \\ v' \end{bmatrix}$, where $v' = D\alpha$ and $\alpha = (\alpha_1, \dots, \alpha_{m+2})^T$. Let $v' = (v'_1, \dots, v'_{l+2})$. Then

$$\|\bar{v}\|^2 = \|v\|^2 + \|v'\|^2 \leq (1 + \delta) + \tau \lfloor \frac{l}{2} \rfloor < 1 + \tau l. \quad (3)$$

The first inequality holds because $v'_1 = v'_2 = 0$, and exactly $\lfloor \frac{l}{2} \rfloor$ of v'_i for $i \geq 3$ are $-\beta$, the rest being zero. The last inequality holds because δ is exponentially small. Also, since v is a non-zero vector, so is \bar{v} , which implies

$$\lambda_1(L^{(D)}) \leq \|\bar{v}\|. \quad (4)$$

We now prove that, assuming a solution to the restricted subset sum instance exists, one such solution can be constructed from an approximate shortest vector.

Let $\bar{w} = \begin{bmatrix} w \\ w' \end{bmatrix}$ be a $(1 + \frac{\tau}{2})$ approximate shortest non-zero vector of $L^{(D)}$, i.e.

$$\lambda_1(L^{(D)})^2 \leq \|\bar{w}\|^2 \leq \left(1 + \frac{\tau}{2}\right) \lambda_1(L^{(D)})^2. \quad (5)$$

We will construct a solution to the subset sum instance, given \bar{w} . Since $\tau = 2/m^\epsilon$, this shows that it is NP-hard to approximate the shortest vector within a factor $\left(1 + \frac{1}{\dim^\epsilon}\right)$, for any constant $\epsilon > 0$, where \dim stands for the dimension of the lattice.

From (3), (4) and (5) we get

$$\|\bar{w}\|^2 \leq \left(1 + \frac{\tau}{2}\right) (1 + \tau l), \quad (6)$$

and by the choice of τ and m ($m^{\epsilon/4} \gg l$), one can show that

$$\|\bar{w}\|^2 \leq 1 + \frac{2}{m^{3\epsilon/4}}.$$

This matches the bound in property 4 of L . Let $w = (w_1, \dots, w_{m+2})$, $w' = (w'_1, \dots, w'_{l+2})$ and $w = \sum_{j=1}^{m+2} \gamma_j v_j$. By property 4, replacing w by $-w$ if necessary, $\gamma_{m+1} = 1$. We now prove that

$$y_i = - \sum_{j \in C_i} \gamma_j$$

is also a solution by showing that, if not, the length of \bar{w} would be too large. It is easy to see that since $\gamma_{m+1} = 1$,

$$w'_1 = \beta l(A - \sum_{i=1}^l a_i y_i),$$

$$w'_2 = \beta l(\lfloor \frac{l}{2} \rfloor - \sum_{i=1}^l y_i),$$

and for $1 \leq j \leq l$,

$$w'_{j+2} = -\beta y_j. \quad (7)$$

Assume the y_i are not a solution. Then, at least one of the following three conditions must hold.

- 1) $\sum_{i=1}^l a_i y_i \neq A$, or
- 2) $\sum_{i=1}^l y_i \neq \lfloor \frac{l}{2} \rfloor$, or
- 3) $\exists i y_i \notin \{0, 1\}$.

If 1) holds, then $|w'_1| \geq \beta l$, which means

$$\|\bar{w}\|^2 = \|w\|^2 + \|w'\|^2 \geq 1 + \beta^2 l^2 = 1 + \tau l^2,$$

where $\|w\| \geq 1$ holds by property 1 of L . This contradicts (6). If 2) holds, then $|w'_2| \geq \beta l$, and we get a similar contradiction again. Finally, it can be shown that if for some i , $y_i \notin \{0, 1\}$ and $\sum_{j=1}^l y_j = \lfloor \frac{l}{2} \rfloor$, then

$$\sum_{j=1}^l y_j^2 \geq \left\lfloor \frac{l}{2} \right\rfloor + 2.$$

This means, by (7) and property 1 of L ,

$$\|\bar{w}\|^2 = \|w\|^2 + \|w'\|^2 \geq 1 + \tau \left(\left\lfloor \frac{l}{2} \right\rfloor + 2 \right).$$

Since $\|\bar{v}\|^2 \leq (1 + \delta) + \tau \lfloor \frac{l}{2} \rfloor$ (see (3)),

$$\|\bar{w}\|^2 - \|\bar{v}\|^2 \geq 2\tau - \delta \geq \tau.$$

Due to our choice of m as a sufficiently large polynomial in l , we have

$$\tau l = \frac{2}{m^\epsilon} l < 1.$$

Thus by (3), $\|\bar{v}\|^2 < 2$, and so

$$\|\bar{w}\|^2 - \|\bar{v}\|^2 > \frac{\tau}{2} \|\bar{v}\|^2.$$

Therefore,

$$\|\bar{w}\|^2 > \left(1 + \frac{\tau}{2}\right) \|\bar{v}\|^2 \geq \left(1 + \frac{\tau}{2}\right) \lambda_1(L^{(D)})^2,$$

which contradicts (5).

This completes the proof of Ajtai's result.

Micciancio's Improvement

With the same basic framework, but using the closest vector problem instead of the restricted subset sum problem, Micciancio [55] got an improved hardness result for the SVP. He showed that it is NP-hard, under randomized reductions, to approximate the SVP to within any constant smaller than $\sqrt{2}$, using the fact that it is NP-hard to approximate the CVP to within any constant. (In fact, it is even NP-hard to do so to within a factor $2^{\log^{1-\epsilon} n}$, for an $\epsilon = o(1)$ [26], but this does not seem to lead to any improvement in his proof.)

To describe this result, it is convenient to formalize the approximation problems as promise problems [29]. The following defines the problem to approximate the closest vector within a factor $c \geq 1$.

CVP Promise Problem

Given an instance (B, y, d) , where $B \in \mathbb{Z}^{n \times k}$ is a basis matrix, $y \in \mathbb{Z}^n$ is a target vector, and $d \in \mathbb{R}$, with the promise that either $\|Bx - y\| \leq d$ for some $x \in \mathbb{Z}^k$, or $\|Bx - y\| > cd$ for all $x \in \mathbb{Z}^k$, decide which is the case.

Arora et.al.[5] showed that for all constants $c \geq 1$, this promise problem is NP-hard. From the proof in [5] one gets that even the following modified version of the above problem is NP-hard for all constants $c \geq 1$.

Modified CVP Promise Problem

Given an instance (B, y, d) , where $B \in \mathbb{Z}^{n \times k}$, $y \in \mathbb{Z}^n$, and $d \in \mathbb{R}$, with the promise that either $\|Bx - y\| \leq d$ for some $x \in \{0, 1\}^k$, or $\|Bx - \alpha y\| > cd$ for all $x \in \mathbb{Z}^k$ and for all $\alpha \in \mathbb{Z} \setminus \{0\}$, decide which is the case.

We will call instances that satisfy the first alternative, YES instances, and those that satisfy the second one, NO instances. Note that, in the modified problem, a YES instance has a 0-1 solution and a NO instance has no solution even for arbitrary integral x and arbitrary (non-zero) multiples of the target vector.

Here is the definition of the corresponding SVP promise problem. It formalizes the problem of approximating the SVP within a factor c' .

SVP Promise Problem

Given an instance (V, t) , where V is a basis matrix, and $t \in \mathbb{R}$, with the promise that either $\|Vw\| \leq t$ for some non-zero integral w , or $\|Vw\| > c't$ for all non-zero integral w , decide which is the case.

We define YES and NO instances in a similar manner.

Micciancio gave a randomized many-one reduction that reduces the modified CVP promise problem with $c = \sqrt{2/\epsilon}$ to the SVP promise problem with $c' = \sqrt{2/(1+2\epsilon)}$, for any constant $\epsilon > 0$, mapping YES instances to YES instances and NO instances to NO instances. This shows that the SVP is NP-hard to approximate within any constant smaller than $\sqrt{2}$.

The heart of his proof is a technical lemma that asserts the existence of a probabilistic algorithm that on input 1^k , where k is from the CVP promise

problem instance, constructs a lattice $L \in \mathbb{R}^{(m+1) \times m}$, a matrix $C \in \mathbb{Z}^{k \times m}$, and an $s \in \mathbb{R}^{m+1}$, such that with high probability,

- For every non-zero $z \in \mathbb{Z}^m$, $\|Lz\|^2 > 2$, and
- For all $x \in \{0, 1\}^k$, $\exists z \in \mathbb{Z}^m$, such that $Cz = x$ and $\|Lz - s\|^2 < 1 + \epsilon$.

Here, m depends polynomially on k .

The lattice L above is essentially the same as Ajtai's lattice L_A and C can be thought of as representing the 0-1 vector x by z . The existence of such a C and the fact that such a C can be randomly constructed depends on a version of Sauer's Lemma.

Let (B, y, d) be a given instance to the CVP promise problem with $c = \sqrt{2/\epsilon}$. The reduction maps it to the instance (V, t) of the SVP promise problem with $c' = \sqrt{2/(1+2\epsilon)}$, where

$$V = \left(\begin{array}{c|c} L & -s \\ \frac{\sqrt{\epsilon}}{d} BC & -\frac{\sqrt{\epsilon}}{d} \cdot y \end{array} \right)$$

and $t = \sqrt{1+2\epsilon}$. Note that $c't = \sqrt{2}$.

Let (B, y, d) be a YES instance. That is, $\|Bx - y\| \leq d$ for some $x \in \{0, 1\}^k$. Then $\exists z \in \mathbb{Z}^m$, such that $\|(BC)z - y\| \leq d$ and $\|Lz - s\|^2 < 1 + \epsilon$. Let w be the vector $\begin{pmatrix} z \\ 1 \end{pmatrix}$. Then

$$\|Vw\|^2 \leq (1 + \epsilon) + \frac{\epsilon}{d^2} \cdot d^2 = 1 + 2\epsilon = t^2.$$

Let (B, y, d) be a NO instance. Let $w = \begin{pmatrix} z \\ \alpha \end{pmatrix}$ be a non-zero vector in \mathbb{Z}^{m+1} , where $z \in \mathbb{Z}^m$ and $\alpha \in \mathbb{Z}$. If $\alpha = 0$, then $z \neq 0$ and so

$$\|Vw\| \geq \|Lz\| > \sqrt{2} = c't.$$

If $\alpha \neq 0$, then

$$\|Vw\| \geq \frac{\sqrt{\epsilon}}{d} \|B(Cz) - \alpha y\| > \frac{\sqrt{\epsilon}}{d} \sqrt{\frac{2}{\epsilon}} d = \sqrt{2} = c't.$$

This completes the description of Micciancio's result.

Other Hardness Results

Dinur, Kindler and Safra [26] have recently improved the hardness factor for CVP. They show that CVP is NP-hard to approximate within a factor $2^{\log^{1-\epsilon} n}$, for an $\epsilon = o(1)$. Blömer and Seifert [10] study two problems considered by Ajtai in his worst-case/ average-case connection. These are the problems of computing a shortest set of independent lattice vectors and a shortest basis. Using the result of [26], they prove that both these problems are hard to approximate within a

factor $n^{c/\log \log n}$, for some constant $c < 1$. Goldreich et al [36] show a reduction from the CVP to the SVP. While this reduction does not give us an improved hardness result, it has the properties of preserving the factor of approximation for the two problems and the dimension of the lattice.

Ravikumar and Sivakumar [61] consider the problem of deciding whether a lattice vector shorter than a given bound exists, under the promise that there is at most one such vector (not counting its negation). They prove a randomized reduction from the decision version of the general shortest vector problem to this problem, in the style of Valiant and Vazirani [64]. Lattice problems for a special kind of lattice defined by certain graphs have been studied in [18].

5 Non-NP-Hardness Results

To what extent can we expect to improve further the approximation factor for SVP and remain NP-hard? The current proof appears not feasible beyond $\sqrt{2}$. On the other hand, the best polynomial time approximation algorithms of Lovász and Schnorr are exponential in the approximation factor.

For polynomially bounded factors, transference theorems provide evidence that beyond a factor of $\Theta(n)$, the approximate SVP is not NP-hard. This is a result of Lagarias, Lenstra and Schnorr [48]. Transference theorems in the Geometry of Numbers give bounds to quantities such λ_i of the primal and the dual lattice. In [48] the following theorem is proved

$$1 \leq \lambda_i(L) \lambda_{n-i+1}(L^*) \leq \frac{1}{6} n^2,$$

for $n \geq 7, 1 \leq i \leq n$. This already gives an “NP proof” for a lower bound for $\lambda_1(L)$ up to a factor of $\Theta(n^2)$ by guessing an appropriate set of linearly independent lattice vectors of L^* all with length at most $\lambda_n(L^*)$.

Lagarias, Lenstra and Schnorr [48] proved more. A basis $\{b_1, b_2, \dots, b_n\}$ is said to be reduced in the sense of Korkin and Zolotarev, if the following hold:

1. $\|b_1\| = \lambda_1(L)$.
2. Let $\{\hat{b}_1, \hat{b}_2, \dots, \hat{b}_n\}$ be the Gram-Schmidt orthogonalization of $\{b_1, \dots, b_n\}$,

$$\hat{b}_i = b_i - \sum_{k < i} \mu_{ik} \hat{b}_k, \quad 1 \leq i \leq n.$$

- Then $|\mu_{ik}| \leq 1/2$, $1 \leq k < i \leq n$.
3. If $L^{(n-i+1)}$ is the orthogonal projection of L to $(\text{lsp}\{b_1, \dots, b_{i-1}\})^\perp$ then $\|\hat{b}_i\| = \lambda_1(L^{(n-i+1)})$.

Essentially, a Korkin-Zolotarev basis is one which is *weakly reduced*, and the orthogonal projection of b_i is a vector of minimum length in the orthogonal projection of L in the complement of $\{b_1, \dots, b_{i-1}\}$. In terms of Lovász’s algorithm, if instead of comparing $b_i(i)$ and $b_{i+1}(i)$, we searched for a vector of minimum

length in $\text{lsp}\{b_i(i), \dots, b_n(i)\}$, and called it b_i , we would have obtained a Korkin-Zolotarev basis. (Of course then this algorithm would have run in exponential time.)

Let B^* be a Korkin-Zolotarev basis of L^* . Its dual basis $B = \{b_1, b_2, \dots, b_n\}$ is called a dual Korkin-Zolotarev basis of L . Let $\lambda(B) = \min\{||\widehat{b}_i|| \mid 1 \leq i \leq n\}$, where $\{\widehat{b}_1, \widehat{b}_2, \dots, \widehat{b}_n\}$ is the Gram-Schmidt orthogonalization of B . Then it is shown in [48] that

$$\lambda(B) \leq \lambda_1(L) \leq n\lambda(B).$$

In particular this gives a way to provide an “NP proof” of a lower bound for $\lambda_1(L)$ up to a factor of n by guessing an appropriate basis B^* and then calculating B . This places the promise problem of approximating $\lambda_1(L)$ up to a factor n within coNP.² Thus if $\text{NP} \neq \text{coNP}$, then approximating $\lambda_1(L)$ up to a factor n is not NP-hard in the sense of Karp reductions. More precisely, if $\text{NP} \neq \text{coNP}$, then there is no deterministic polynomial time reduction σ from SAT, $\sigma(\varphi) = (L, \lambda)$, such that if $\varphi \in \text{SAT}$, then $\lambda_1(L) \leq \lambda$, and if $\varphi \notin \text{SAT}$, then $\lambda_1(L) \geq n\lambda$.

Theorem 6 (Lagarias, Lenstra, Schnorr). *If $\text{NP} \neq \text{coNP}$, then the problem of approximating $\lambda_1(L)$ within a factor n is not NP-hard.*

The interplay between the primal and dual lattices and the related transference theorems play important roles in Ajtai’s worst-case to average-case connection as well. We will discuss this topic in more detail in the next section. Here we present the following rather pretty result due to Goldreich and Goldwasser which improved the approximation factor for non-NP-hardness to \sqrt{n} .

The proof of Goldreich and Goldwasser [32] is based on constant round interactive proof systems. More precisely, they give a bounded round interactive proof system for proving a lower bound up to a factor \sqrt{n} for both SVP as well as CVP. Of course the number of rounds can be reduced to one, either by standard techniques or by directly parallelizing their IP protocol. Also by standard techniques private coins can be replaced by public coins, so that what they showed can be stated as follows:

Theorem 7 (Goldreich, Goldwasser). *The problem of approximating $\lambda_1(L)$ within a factor \sqrt{n} is in $\text{NP} \cap \text{coAM}$. Thus if this problem is NP-hard under Karp reductions in the sense given above, then $\Sigma_2^p = \Pi_2^p$.*

The last statement follows from a well-known result of Bopanna et. al. [12] which states that if $\text{coNP} \subseteq \text{AM}$, then $\Sigma_2^p = \Pi_2^p$.

The restriction to Karp reductions has been improved recently to general Cook reductions to promise problems in this result [22].

² Of course, technically a promise problem is not a decision problem while coNP is a decision problem class. But the meaning of this is clear and one can always modify the definitions slightly to make it proper.

The basic idea of the IP protocol of [32] is rather simple and elegant and we will describe it here.

Suppose L satisfies the promise of either $\lambda_1(L) \leq t$ or $\lambda_1(L) > t \cdot \sqrt{n}$, and the prover claims that $\lambda_1(L) > t \cdot \sqrt{n}$. Imagine we surround each lattice point $p \in L$ a ball $B_p(r)$ centered at p with radius $r = t \cdot \sqrt{n}/2$. If the prover P is correct, then all such balls are disjoint. Now the verifier randomly picks a lattice point p in secret, and randomly picks a point z in $B_p(r)$. The verifier presents z to the prover, who should respond with p , the center of the ball from which z was chosen. It is clear that for an honest prover P with unlimited computing power, since all the balls $B_p(r)$ are disjoint, he has no difficulty meeting his obligation. However, suppose the prover P' is dishonest, so that in fact $\lambda_1(L) \leq t$. Then for any lattice point p picked by the verifier, there is at least one nearby lattice point p' with $\|p - p'\| \leq t$. Then $B_p(r)$ and $B_{p'}(r)$ would have a large intersection. This follows from the fact that the radius is almost $n^{1/2}$ times the distance of their respective centers. It follows that there is a significant probability that a dishonest prover will be caught, since in case a point $z \in B_p(r) \cap B_{p'}(r)$ is chosen, the verifier could equally have chosen p or p' .

The exponent $1/2$ in this interactive proof protocol comes from the well known fact that in n -dimensional space, two unit balls with center distance d have a significant intersection if $d < 1/\sqrt{n}$, and a negligible intersection if $d > 1/n^{1/2-\epsilon}$, for any $\epsilon > 0$. With some care the proof in [32] can improve the factor \sqrt{n} to $\sqrt{n}/\log n$. It also shows the same bound for the Closest Vector Problem.

What about some other problems? The problem of n^c -unique shortest vector problem is prominent in the Ajtai worst-case to average-case connection. It also plays an important role in the Ajtai-Dwork public-key cryptosystem. Recall that a lattice is said to have an n^c -unique shortest vector if $\lambda_2(L)/\lambda_1(L) \geq n^c$. Equivalently, there exists $v \in L$, $v \neq 0$, such that for all $v' \in L$, if $\|v'\| < n^c \cdot \|v\|$, then v' is an integral multiple of v .

Define the following promise problem:

The n^c -Unique Shortest Lattice Vector Problem:

Given a lattice with a n^c -unique shortest vector v , find the shortest vector $\pm v$.

Building on the idea of Goldreich and Goldwasser [32], Cai [16] proved the following:

Theorem 8. *The n^c -unique shortest lattice vector problem for $c \leq 1/4$ is not NP-hard under Karp reductions unless the polynomial time hierarchy collapses to $\Sigma_2^p = \Pi_2^p$.*

It is not yet clear whether Theorem 8 can be improved to hold for general Cook reductions as well.

6 Transference Theorems

We have already mentioned the transference theorem of Lagarias, Lenstra and Schnorr [48] in the last section. There is a long history in geometry of numbers

to study relationships between various quantities such as the successive minima associated with the primal and dual lattices, L and L^* . Such theorems are called transference theorems. The estimate for the product

$$\lambda_i(L)\lambda_{n-i+1}(L^*)$$

has a illustrious history: Mahler [54] proved that the upper bound $(n!)^2$ holds for all lattices. This was improved by Cassels [24] to $n!$. The first polynomial upper bound was obtained by Lagarias, Lenstra and Schnorr [48] as mentioned. The best estimate for this product is due to Banaszczyk [9], who showed that

$$1 \leq \lambda_i(L)\lambda_{n-i+1}(L^*) \leq Cn,$$

for some universal constant C . The Banaszczyk bound is optimal up to a constant, for Conway and Thompson (see [56]) showed that there exists a self-dual lattice family $\{L_n\}$ with $\lambda_1(L_n) = \Omega(\sqrt{n})$.

Part 2) and part 3) of Ajtai's worst-case to average-case connection in Theorem 3 are proved via transference type argument. Basically, if one can get a good estimate for the basis length for any lattice, one can apply this to the dual L^* . From a good estimate for $\text{bl}(L^*)$, thus $\lambda_n(L^*)$, a transference theorem gives estimate for $\lambda_1(L)$. This is part 2) in Theorem 3. Part 3) employs some additional argument also of a transference type. We will discuss these matters in more detail. But first we take a closer look at transference theorems.

In addition to λ_i , there are several other lattice quantities that have been studied. The covering radius of L is defined to be the minimum radius of balls centered at each lattice point whose union covers \mathbb{R}^n .

$$\mu(L) = \min\{r \mid L + B(0; r) = \mathbb{R}^n\}.$$

Also if $d(u, L)$ denotes the minimum distance from a point u in \mathbb{R}^n to a point in L , then

$$\mu(L) = \max\{d(u, L) \mid u \in \mathbb{R}^n\}.$$

(The minimum and maximum are obviously achieved.)

We have seen the quantity

$$\xi = \sup_L \max_{1 \leq i \leq n} \lambda_i(L)\lambda_{n-i+1}(L^*),$$

where the supremum is taken over all n -dimensional lattices. Regarding covering radius $\mu(L)$ the relevant quantity is

$$\eta = \sup_L \mu(L)\lambda_1(L^*).$$

By triangle inequality $\mu(L) \leq \frac{1}{2}n\lambda_n(L)$, so that

$$\eta \leq \frac{1}{2}n\xi.$$

Given any L , we say a sublattice $L' \subseteq L$ is a *saturated sublattice* if $L' = L \cap \Pi$, where Π is the linear subspace of \mathbb{R}^n spanned by L' . Saturated sublattices of dimension $n - 1$ are in one-to-one correspondence with primitive vectors of L^* . (A lattice vector $v \neq 0$ is primitive if it is not an integral multiple of any other vector in the lattice except $\pm v$.) The correspondence is simply $L' = L \cap \{v\}^\perp$ and $\{v\}^\perp = \text{lsp}(L')$. For any L and a saturated sublattice L' of dimension $n - 1$ with normal (and primitive) vector $v \in L^*$, L is a disjoint union of parallel translations of L' ,

$$L = \bigcup_{k \in \mathbb{Z}} (L' + ku),$$

for some $u \in L$ such that $\langle u, v \rangle = 1$. Thus, each pair of nearest hyperplanes $\{v\}^\perp + ku$ and $\{v\}^\perp + (k+1)u$ has orthogonal distance $\langle u, \frac{v}{||v||} \rangle = \frac{1}{||v||}$. We call this a parallel decomposition of L .

For any L and any $u \in \mathbb{R}^n \setminus L$, we can compare $d(u, L)$, to the distance from u to the closest parallel translation of some $\{v\}^\perp = \text{lsp}(L')$ which intersects L , over all such L' . Let

$$d_{\mathbb{Z}}(\langle u, v \rangle) = |\langle u, v \rangle - \lceil \langle u, v \rangle \rceil|$$

be the fractional part of $\langle u, v \rangle$ rounded to the nearest integer, then we consider

$$\delta = \sup_{v \in L^*, \langle u, v \rangle \notin \mathbb{Z}} \frac{d_{\mathbb{Z}}(\langle u, v \rangle)}{||v||},$$

which measures the distance from u to the closest parallel translation, maximized among all directions $v \in L^*$. Now the following quantity is defined

$$\zeta = \sup_L \sup_{u \in \mathbb{R}^n \setminus L} \frac{d(u, L)}{\delta}.$$

By definition $d_{\mathbb{Z}}(\langle u, v \rangle) \leq 1/2$ and $||v|| \geq \lambda_1(L^*)$, so that $\delta \leq \frac{1}{2\lambda_1(L^*)}$. Hence

$$\zeta \geq 2\eta.$$

An upper bound $\zeta \leq \beta$ says that $\forall L$ and $\forall u \notin L$, there exists a parallel decomposition where the distance from u to the nearest lattice hyperplane is $\geq \beta d(u, L)$.

Lagarias et al. [48] proved that $\xi \leq \frac{1}{6}n^2$ and $\eta \leq \frac{1}{2}n^{3/2}$. Babai [6] proved that $\zeta \leq Cn$ for some universal constant C . Håstad [40] showed that $\zeta \leq 6n^2 + 1$. Similar bounds for ξ , η and ζ were also shown by Banaszczyk [8]. The best bounds for ξ , η and ζ were shown later by Banaszczyk [9], where ξ , η and ζ are all bounded by $O(n)$. The Banaszczyk bounds are all optimal up to a constant by the Conway-Thompson family of lattices (see [56]).

In [14] an extension of Banaszczyk's theorem of [9] is proved. Define $g_i(L)$ to be the minimum r such that the sublattice generated by $L \cap B(0; r)$ contains an i -dimensional saturated sublattice L' , where $1 \leq i \leq n$. When $i = n$, it is called

the *generating radius* and is denoted by $g(L)$. Clearly $g(L)$ is the minimum r such that a ball $B(0; r)$ centered at 0 with radius r contains a set of lattice vectors generating L . The study of $g(L)$ is motivated by the investigation of $\text{bl}(L)$ and its relation to $\lambda_n(L)$. Clearly

$$\lambda_n(L) \leq g(L) \leq \text{bl}(L).$$

The following inequality is shown in [14] for every lattice L of dimension n , using and extending the techniques of [9]:

$$g_i(L) \cdot \lambda_{n-i+1}(L^*) \leq Cn, \quad (8)$$

for some universal constant C , and for all i , $1 \leq i \leq n$. We will sketch the proof for the case $i = n$ for the generating radius $g(L)$.

The main tools of the proof are Gaussian-like measures on a lattice, and their Fourier transforms. For a given lattice L we define

$$\sigma_L(\{v\}) = \frac{e^{-\pi||v||^2}}{\sum_{x \in L} e^{-\pi||x||^2}}. \quad (9)$$

The Fourier transform of σ_L is

$$\widehat{\sigma_L}(u) = \int_{x \in \mathbb{R}^n} e^{2\pi i \langle u, x \rangle} d\sigma_L = \sum_{v \in L} e^{2\pi i \langle u, v \rangle} \sigma_L(\{v\}), \quad (10)$$

where $u \in \mathbb{R}^n$. Note that σ_L is an even function, so that

$$\widehat{\sigma_L}(u) = \sum_{v \in L} \sigma_L(\{v\}) \cos(2\pi \langle u, v \rangle). \quad (11)$$

Define

$$\tau_L(u) = \frac{\sum_{y \in L+u} e^{-\pi||y||^2}}{\sum_{x \in L} e^{-\pi||x||^2}}. \quad (12)$$

Then the following identity holds

Lemma 4.

$$\widehat{\sigma_L}(u) = \tau_{L^*}(u). \quad (13)$$

The proof of Lemma 4 uses Poisson summation formula, see [42,9]. The following lemma is proved in [9] and is crucial:

Lemma 5. *For each $c \geq 1/\sqrt{2\pi}$,*

$$\sigma_L(L \setminus B(0; c\sqrt{n})) < \left(c\sqrt{2\pi}ee^{-\pi c^2}\right)^n, \quad (14)$$

and for all $u \in \mathbb{R}^n$,

$$\frac{\sum_{v \in (L+u) \setminus B(0; c\sqrt{n})} e^{-\pi||v||^2}}{\sum_{x \in L} e^{-\pi||x||^2}} < 2 \left(c\sqrt{2\pi}ee^{-\pi c^2}\right)^n, \quad (15)$$

where $B(0; c_1\sqrt{n})$ is the n -dimensional ball of radius $c_1\sqrt{n}$ centered at 0.

This lemma basically says that the total weight under σ_L of all lattice (or affine lattice) points outside of radius $c\sqrt{n}$ is exponentially small.

Now we prove (8) for $i = n$ and $C = 3/(2\pi)$. Suppose $g(L)\lambda_1(L^*) > 3n/2\pi$. Let c_1 and c_2 be two constants, such that $c_1c_2 > 3/2\pi$ and $c_1 > 1/\sqrt{2\pi}$ and $c_2 > 3/\sqrt{2\pi}$. By substituting L with sL for a suitable scaling factor s , we may assume that

$$g(L) > c_1\sqrt{n} \quad \text{and} \quad \lambda_1(L^*) > c_2\sqrt{n}.$$

Let L' be the sublattice of L generated by the intersection $L \cap B(0; c_1\sqrt{n})$. Then L' is a proper sublattice of L , since $g(L) > c_1\sqrt{n}$. If $\dim L' < n$, then let P be the linear span of L' , and let b_1, \dots, b_i be a lattice basis of $L \cap P$, where $i = \dim L' < n$. This can be extended to a lattice basis $b_1, \dots, b_i, \dots, b_n$ for L and we may replace L' by the sublattice generated by $b_1, \dots, b_i, \dots, 2b_n$, say. Thus without loss of generality we may assume L' is of dimension n . The important point is that we have a proper sublattice $L' \subset L$, which is of dimension n and contains $L \cap B(0; c_1\sqrt{n})$.

For any fixed $u \in \mathbb{R}^n$,

$$\begin{aligned} \widehat{\sigma_L}(u) &= \sum_{v \in L} \sigma_L(\{v\}) \cos(2\pi\langle u, v \rangle) \\ &= \sum_{v \in L'} \sigma_{L'}(\{v\}) \cos(2\pi\langle u, v \rangle) \\ &\quad + \sum_{v \in L' \setminus L} (\sigma_L(\{v\}) - \sigma_{L'}(\{v\})) \cos(2\pi\langle u, v \rangle) \\ &\quad + \sum_{v \in L \setminus L'} \sigma_L(\{v\}) \cos(2\pi\langle u, v \rangle) \\ &= \widehat{\sigma_{L'}}(u) + A + B, \quad \text{say.} \end{aligned}$$

Since $L \cap B(0; c_1\sqrt{n}) \subset L'$, the last term

$$\begin{aligned} |B| &\leq \sum_{v \in L \setminus B(0; c_1\sqrt{n})} \sigma_L(\{v\}) \\ &< \left(c_1\sqrt{2\pi}e^{-\pi c_1^2}\right)^n, \end{aligned}$$

by Lemma 5 inequality (14). Denote the last term by ϵ_1^n , say.

For the other error term A , we can show similarly that

$$|A| < \epsilon_1^n.$$

Hence

$$\widehat{\sigma_L}(u) > \widehat{\sigma_{L'}}(u) - 2\epsilon_1^n. \tag{16}$$

Our next task is to show that we can choose an appropriate u so that $\widehat{\sigma_L}(u)$ is small yet $\widehat{\sigma_{L'}}(u)$ is large. By Lemma 4, we have $\widehat{\sigma_L}(u) = \tau_{L^*}(u)$, and $\widehat{\sigma_{L'}}(u) =$

$\tau_{(L')^*}(u)$. Thus we only need to choose a u such that $\tau_{L^*}(u)$ is small and $\tau_{(L')^*}(u)$ is large.

The following lemma is proved in [14].

Lemma 6. *Suppose L_1 is a proper sublattice of L_2 , then there exists a $p \in L_2$, such that*

$$\min_{q \in L_1} \|p - q\| \geq \frac{\lambda_1(L_1)}{3}.$$

(Since a lattice is a discrete subset of \mathbb{R}^n , the above minimum over q clearly exists.)

Now we note that since L' is a full ranked proper sublattice of L , L^* is a proper sublattice of $(L')^*$. That it is proper follows from the identity of index

$$\det((L')^*)/\det(L^*) = \det(L)/\det(L') > 1.$$

By Lemma 6, take a $u \in (L')^*$, such that $\min_{q \in L^*} \|u - q\| \geq \frac{\lambda_1(L^*)}{3}$. Then since $u \in (L')^*$, we have $(L')^* + u = (L')^*$, and

$$\tau_{(L')^*}(u) = \frac{\sum_{x \in (L')^* + u} e^{-\pi\|x\|^2}}{\sum_{x \in (L')^*} e^{-\pi\|x\|^2}} = 1.$$

On the other hand, since

$$\min_{q \in L^*} \|u - q\| \geq \frac{\lambda_1(L^*)}{3} > \frac{c_2}{3}\sqrt{n},$$

we note that no point in $L^* + u$ is within $\frac{c_2}{3}\sqrt{n}$ in norm, and so

$$\begin{aligned} \tau_{L^*}(u) &= \frac{\sum_{x \in L^* + u} e^{-\pi\|x\|^2}}{\sum_{x \in L^*} e^{-\pi\|x\|^2}} \\ &< 2 \left(\frac{c_2}{3} \sqrt{2\pi e} e^{-\pi(\frac{c_2}{3})^2} \right)^n = 2\epsilon_2^n \quad \text{say,} \end{aligned}$$

by Lemma 5 inequality (15). Since both c_1 and $c_2/3 > 1/\sqrt{2\pi}$, we have both ϵ_1 and $\epsilon_2 < 1$ by elementary estimate. Thus it follows from (16) that

$$2\epsilon_2^n > 1 - 2\epsilon_1^n,$$

which is a contradiction for large n .

For the special class of lattices possessing n^ϵ -unique shortest vector, a stronger bound is proved [15], which lead to a further improvement in the Ajtai connection factors of part 2) and 3) in Theorem 3.

Theorem 9. *For every lattice L of dimension n , if L^* has an n^c -unique shortest vector, then*

$$1 \leq \lambda_n(L)\lambda_1(L^*) \leq O(n^\delta),$$

where

$$\delta = \begin{cases} 1 - c & \text{if } 0 < c \leq 1/2, \\ 1/2 & \text{if } 1/2 < c \leq 1, \\ 3/2 - c & \text{if } 1 < c \leq 3/2, \\ 0 & \text{if } c > 3/2. \end{cases}$$

In terms of the Ajtai connection factors in Theorem 3—in part 2) and part 3)—these new transference theorems improve all the factors to the range of approximately 3 and 4. Details can be found in [15]. Here we outline the general idea to derive parts 2) and 3) from 1).

The idea for the estimation of $\lambda_1(L)$ is relatively straightforward. From an estimate of the maximum length of a set of linearly independent vectors from L^* , one gets an estimate of $\lambda_1(L)$, via transference theorem.

To actually compute the shortest vector, the following idea is due to Ajtai [1]. If L^* has an n^c -unique shortest vector v , then L admits a parallel decomposition

$$L = \bigcup_{k \in \mathbb{Z}} (L' + ku),$$

where the parallel hyperplanes containing $L' + ku$ have orthogonal distance much larger than the basis length of L' . Now randomly sample a large polynomial number of lattice points within a certain bound. A $1/n^{O(1)}$ fraction of samples fall on the same parallel hyperplane, and the difference vector of such a pair belongs to the hyperplane $\text{lsp}(L')$. If we can distinguish such pairs from the rest, then we can identify the normal vector for the hyperplane $\text{lsp}(L')$, and by taking out the gcd, we can recover the shortest vector $\pm v$.

For two sample lattice points x and y , if they belong to the same parallel hyperplane, then by including a small fractional vector $(x - y)/N$ to the generating set of L , one does not change $\text{bl}(L)$, since this is controlled by the distance between the parallel hyperplanes.

But if x and y belong to different parallel hyperplanes, then by including $(x - y)/N$ to the generating set of L , the new lattice will have many additional parallel translations of L' between any two originally adjacent parallel hyperplanes $\text{lsp}(L') + ku$ and $\text{lsp}(L') + (k - 1)u$. This will reduce the basis length significantly.

Thus to be able to compute a good estimate of the basis length for L (actually an estimate of $\lambda_n(L)$ will do) leads to the identification of the unique shortest vector for L^* . Clearly improved transference theorem bounds sharpen the provable estimates in Ajtai's worst-case to average-case connection factors.

Acknowledgements

I thank the organizers for the ANTS conference for inviting me to give this Invited Talk. I wish to thank Allan Borodin, Steve Cook and Charlie Rackoff for their hospitality during my recent stay at the University of Toronto. I thank Laci

Lovász and Herb Scarf who introduced me to the beauty of lattice problems. I thank Miki Ajtai, Tom Cusick, Alan Frieze, Oded Goldreich, Shafi Goldwasser, Ravi Kannan, Janos Komlos, Jeff Lagarias, Ajay Nerurkar, Endre Szemerédi and Andy Yao for interesting discussions. I especially thank Ajay Nerurkar for his valuable assistance to me in the preparation of this article.

References

1. M. Ajtai. Generating hard instances of lattice problems. In *Proc. 28th ACM Symposium on the Theory of Computing*, 1996, 99–108. Full version available from ECCC as TR96-007.
2. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. 30th ACM Symposium on the Theory of Computing*, 1998, 10–19. Full version available from ECCC as TR97-047.
3. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th ACM Symposium on the Theory of Computing*, 1997, 284–293. Full version available from ECCC as TR96-065.
4. N. Alon and J. Spencer. The Probabilistic Method (with an appendix on open problems by Paul Erdős). Wiley, 1992.
5. S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Proc. 34th IEEE Symposium on Foundations of Computer Science*, 1993, 724–733.
6. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
7. K. Ball. Cube slicing in \mathbf{R}^n . *Proceedings of the American Mathematical Society*, 97(3):465–473, 1986.
8. W. Banaszczyk. Polar Lattices from the point of view of nuclear spaces. *Rev. Mat. Univ. Complutense Madr.* 2 (special issue):35–46, 1989.
9. W. Banaszczyk. New Bounds in Some Transference Theorems in the Geometry of Numbers. *Mathematische Annalen*, 296:625–635, 1993.
10. J. Blömer and J-P. Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. *Proc. 31st ACM Symposium on Theory of Computing*, pp711–720, 1999.
11. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. Lecture Notes in Computer Science, 1109 (1996), 129–142.
12. R. Boppana, J. Håstad and S. Zachos. Does Co-NP have Short Interactive Proofs? *Information Processing Letters*, 25:127–132, 1987.
13. J-Y. Cai. Some recent progress on the complexity of lattice problems. Available as TR99-006 at <http://www.eccc.uni-trier.de/eccc/>.
14. J-Y. Cai. A New Transference Theorem in the Geometry of Numbers. *The 5th International Computing and Combinatorics Conference*, 113–122, (COCOON) 1999, Tokyo, Japan. Lecture Notes in Computer Science, 1627.
15. J-Y. Cai. Applications of a New Transference Theorem to Ajtai's Connection Factor. In the Proceedings of the 14th Annual IEEE Conference on Computational Complexity, pp 205–214, 1999.
16. J-Y. Cai. A Relation of Primal-Dual Lattices and the Complexity of Shortest Lattice Vector Problem. *Theoretical Computer Science* 207:105–116, 1998.

17. J-Y. Cai and T. Cusick. A Lattice-Based Public-Key Cryptosystem. *Information and Computation* **151**, 17–31, 1999.
18. J-Y. Cai, G. Havas, B. Mans, A. Nerurkar, J-P. Seifert and I. Shparlinski. On routing in circulant graphs. *The 5th International Computing and Combinatorics Conference*, 360–369, (COCOON) 1999, Tokyo, Japan. Lecture Notes in Computer Science, 1627.
19. J-Y. Cai and A. Nerurkar. An Improved Worst-Case to Average-Case Connection for Lattice Problems. In *Proc. 38th IEEE Symposium on Foundations of Computer Science*, 1997, 468–477.
20. J-Y. Cai and A. Nerurkar. Approximating the SVP to within a factor $(1 + \frac{1}{\dim^c})$ is NP-hard under randomized reductions. In *Proc. of the 13th IEEE Conference on Computational Complexity*, 1998, 46–55.
21. J-Y. Cai, A. Pavan and D. Sivakumar. On the Hardness of Permanent. In *Proc. of the 16th International Symposium on Theoretical Aspects of Computer Science*, 1999.
22. J-Y. Cai and A. Nerurkar. A note on the non-NP-hardness of approximate lattice problems under general Cook reductions. Submitted to *Information Processing Letters*.
23. J-Y. Cai. A Worst-Case to Average-Case Connection for CVP. Manuscript. To appear.
24. J. W. S. Cassels. *An Introduction to the Geometry of Numbers*. Berlin Göttingen Heidelberg: Springer 1959.
25. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10:233–260, 1997.
26. I. Dinur, G. Kindler and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *Proc. 39th IEEE Symposium on Foundations of Computer Science*, 1998, 99–109.
27. P. G. L. Dirichlet. Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die Reine und Angewandte Mathematik*, 40:209–227, 1850.
28. A. Dupré. Sur le nombre de divisions à effectuer pour obtenir le plus grande commun diviseur entre deux nombres entiers. *Journal de Mathématiques*, 11:41–74, 1846.
29. S. Even, A. L. Selman and Y. Yacobi. The Complexity of Promise Problems with Applications to Public-key Cryptography. *Information and Control* 61:159–173, 1984.
30. U. Feige and C. Lund. On the hardness of computing permanent of random matrices. In *Proc. 14th ACM Symposium on Theory of Computing*, 1982, 643–654.
31. P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43:169–174, 1992.
32. O. Goldreich and S. Goldwasser. On the Limits of Non-Approximability of Lattice Problems. In *Proc. 30th ACM Symposium on Theory of Computing*, 1998, 1–9.
33. O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. Available from ECCC as TR96-042.
34. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology – CRYPTO ’97*, Burton S. Kaliski Jr. (Ed.), Lecture Notes in Computer Science, 1294:112–131, Springer-Verlag, 1997.
35. O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In *Advances in Cryptology – CRYPTO ’97*, Burton S. Kaliski Jr. (Ed.), Lecture Notes in Computer Science, 1294:105–111, Springer-Verlag, 1997.

36. O. Goldreich, D. Micciancio, S. Safra and J-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. Available from ECCC as TR99-002.
37. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, 1988.
38. O. Goldreich and D. Ron and M. Sudan. Chinese remaindering with errors. Available from ECCC as TR 98-062.
39. P. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers*. North-Holland, 1987.
40. J. Håstad. Dual Vectors and Lower Bounds for the Nearest Lattice Point Problem. *Combinatorica*, 8:75–81, 1988.
41. C. Hermite. Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objets de la théorie des nombres. *Journal für die Reine und Angewandte Mathematik*, 40:261–278, 279–290, 291–307, 308–315, 1850.
42. E. Hewitt and K. A. Ross. *Abstract Harmonic Analysis*, Vol II. Berlin Göttingen Heidelberg: Springer 1970.
43. E. Kaltofen. Polynomial factorization 1987–1991. *LATIN '92*, I. Simon (Ed.), Lecture Notes in Computer Science, 583:294–313, Springer, 1992.
44. R. Kannan. Minkowski's convex body theory and integer programming. *Mathematics of Operations Research*, 12:415–440, 1987.
45. N. Koblitz. *Introduction to Elliptic Curves and Modular Forms*. Springer-Verlag, GTM 97, 1984.
46. A. Korkin and G. Zolotarev. Sur les formes quadratiques positives quaternaires. *Mathematische Annalen*, 5:581–583, 1872.
47. J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM Journal of Computing*, 14:196–209, 1985.
48. J. C. Lagarias, H. W. Lenstra, and C. P. Schnorr. Korkin-Zolotarev Bases and Successive Minima of a Lattice and its Reciprocal Lattice. *Combinatorica*, 10(4):333–348, 1990.
49. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. In *Proc. 24th IEEE Symposium on Foundations of Computer Science*, 1983, 1 – 10.
50. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.
51. H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.
52. L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*. SIAM, Philadelphia, 1986.
53. L. Lovász and H. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17(3):751–764, 1992.
54. K. Mahler. Ein Übertragungsprinzip für konvexe Körper. *Čas. Pěstování Mat. Fys.* 68:93–102, 1939.
55. D. Micciancio. The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant. In *Proc. 39th IEEE Symposium on Foundations of Computer Science*, 1998, 92–98.
56. J. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Berlin Heidelberg New York: Springer 1973.
57. H. Minkowski. Über die positiven quadratischen Formen und über kettenbruchähnliche Algorithmen. *Crelles Journal für die Reine und Angewandte Mathematik*, 107:278–297, 1891.
58. P. Nguyen and J. Stern. A converse to the Ajtai-Dwork security proof and its cryptographic implications. Available from ECCC as TR98-010.

59. P. Nguyen and J. Stern. Lattice Reduction in Cryptology: An Update. In these proceedings.
60. A. Odlyzko and H.J.J. te Riele. Disproof of the Mertens conjecture. *Journal für die Reine und Angewandte Mathematik*, 357:138–160, 1985.
61. S. Ravikumar and D. Sivakumar. A note on the shortest lattice vector problem. In the Proceedings of the 14th Annual IEEE Conference on Computational Complexity, pp 200–204, 1999.
62. C. P. Schnorr. A hierarchy of polynomial time basis reduction algorithms. *Theory of Algorithms*, 375–386, 1985.
63. C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.
64. L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.
65. B. Vallée. Un problème central en géométrie algorithmique des nombres: la réduction des réseaux;atour de l'algorithme LLL. *Inform. Théor. Appl.*, 345–376, 1989. English transl. by E. Kranakis, *CWI Quart* 3:95–120, 1990.
66. P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in lattices. Technical Report 81-04, Mathematics Department, University of Amsterdam, 1981.

Rational Points Near Curves and Small Nonzero $|x^3 - y^2|$ via Lattice Reduction

Noam D. Elkies

Department of Mathematics, Harvard University
Cambridge, MA 02138 USA
`elkies@math.harvard.edu`

Abstract. We give a new algorithm using linear approximation and lattice reduction to efficiently calculate all rational points of small height near a given plane curve C . For instance, when C is the Fermat cubic, we find all integer solutions of $|x^3 + y^3 - z^3| < M$ with $0 < x \leq y < z < N$ in heuristic time $\ll (\log^{O(1)} N)M$ provided $M \gg N$, using only $O(\log N)$ space. Since the number of solutions should be asymptotically proportional to $M \log N$ (as long as $M < N^3$), the computational costs are essentially as low as possible. Moreover the algorithm readily parallelizes. It not only yields new numerical examples but leads to theoretical results, difficult open questions, and natural generalizations. We also adapt our algorithm to investigate Hall's conjecture: we find all integer solutions of $0 < |x^3 - y^2| \ll x^{1/2}$ with $x < X$ in time $O(X^{1/2} \log^{O(1)} X)$. By implementing this algorithm with $X = 10^{18}$ we shattered the previous record for $x^{1/2}/|x^3 - y^2|$. The $O(X^{1/2} \log^{O(1)} X)$ bound is rigorous; its proof also yields new estimates on the distribution mod 1 of $(cx)^{3/2}$ for any positive rational c .

1 Introduction

One intriguing class of Diophantine problem concerns small values of homogeneous polynomials. In the simplest nontrivial case of a polynomial in three variables defining a projective plane curve $C : P(X, Y, Z) = 0$, the problem can be reformulated thus: given a plane curve C , describe for each positive N, δ the rational points of height at most N in \mathbb{P}^2 which are at distance at most δ of C . With present-day methods, hardly any nontrivial results can be proved on the number or existence of such points. But one can still seek numerical evidence, and efficient algorithms for obtaining this evidence. The direct approach is to try all x, y with $|x|, |y| \leq N$, and for each pair to solve $P(x, y, z) = 0$ for $z \in [-N, N]$, recording those cases in which z is sufficiently close to an integer. This requires space $O(\log(N))$ but time $(N^2 + \delta N^3) \log^{O(1)} N$, which is inefficient once δ is much smaller than N^{-1} since for general $C, N, \delta \gg N^{-3}$ the number of solutions should be proportional to δN^3 . We give a new algorithm, also requiring only $O(\log(N))$ space, but with heuristic running time $(N + \delta N^3) \log^{O(1)} N$. Thus as long as $\delta \gg N^{-2}$ we expect to find all the points of height $\leq N$ and distance $\leq \delta$ in time only $\log^{O(1)} N$ per point. Moreover, our method readily parallelizes, since it divides the computation into many independent subproblems.

We describe this algorithm, give the heuristic estimate for its run time, and briefly discuss the problem, which seems quite difficult, of proving our heuristic time estimates. We prove (Thm.1) that an alternative description of those points can always be computed in the heuristically expected time. We then discuss natural generalizations to other valuations and higher dimensions.

An algorithm for finding rational points *near* a variety can in particular find rational points *on* the variety; applying our methods to embeddings of the variety in projective spaces of high dimension we obtain a new approach to this fundamental problem in computational number theory which improves on existing methods in several important cases. This approach also works for non-algebraic varieties, and even yields a theoretical result (Thm.4) on the paucity of rational points on non-algebraic analytic curves.

We next describe experimental results of the implementation of our algorithm to various curves of interest, notably the Fermat curves of degree $n > 2$, where some of our experimental findings led us to new polynomial families of small values of $|z^n - y^n - x^n|$ (Thm.5). We devote a separate section to the case of the cubic Fermat curve, corresponding to small values of $|z^3 - y^3 - x^3|$, a problem for which there is already some literature and the heuristics are subtler. In particular, we found for several integers $d < 10^3$ the first representation of d as a sum of three integer cubes; D.J.Bernstein has since extended the search up to $N = 2 \cdot 10^9$ and beyond, and found many new solutions, including one for $d = 30$ which was a long-standing open problem.

Finally we show how to modify our algorithm to efficiently search for small nonzero values of $|x^3 - y^2|$. This is the topic of Hall's conjecture, which is part of a web of important Diophantine problems surrounding the ABC conjecture of Masser and Oesterlé. The conjecture asserts that $x^3 - y^2$ is either zero or $\gg_\epsilon x^{1/2-\epsilon}$ for all $x, y \in \mathbb{Z}$. We are able to find all solutions of $0 < |x^3 - y^2| \ll x^{1/2}$ with $x \leq X$ in time $O(X^{1/2} \log^{O(1)} X)$, again using only $O(\log X)$ space. Using this improvement on the obvious $X \log^{O(1)} X$ method of trying all $x \leq X$, we computed all cases of $0 < |x^3 - y^2| < x^{1/2}$ with $X \leq 10^{18}$. We found ten new solutions, including most notably

$$5853886516781223^3 - 447884928428402042307918^2 = 1641843$$

with $x^{1/2}/|x^3 - y^2| = 46.600+$, improving the previous record by a factor of almost 10. In this case the time estimate is *not* heuristic; its proof not only streamlined the computation but even yields new theorems on the distribution mod 1 of $(cx)^{3/2}$ for any positive rational c . We announce some of these results at the end of the present paper; the full statements and proofs will appear elsewhere.

2 The Algorithm in Theory

2.1 Specification and Heuristic Analysis

While we are mainly interested in algebraic plane curves C , the algorithm does not require so strong a hypothesis: we can find¹ $2063^\pi + 8093^\pi - 8128^\pi = 0.019369 -$ as well as $386692^7 + 411413^7 = (1 - 1.035 \dots 10^{-18})441849^7$. All we need is that C is the image of a differentiable map $\phi : [0, 1] \rightarrow \mathbb{RP}^2$ with bounded second derivatives. Fix a positive $\delta \leq 1$, and assume $\delta \gg N^{-2}$ for reasons given in the next paragraph. Partition $[0, 1]$ into $O(\delta^{-1/2})$ intervals I_m each of length $|I_m| = O(\delta^{1/2})$. On each I_m , approximate ϕ to within $O(|I_m|^2) = O(\delta)$ by a linear approximation $\bar{\phi}$. Then a point at distance $\leq \delta$ from $\phi(I_m)$ remains at distance $\ll \delta$ from $\bar{\phi}(I_m)$.

We now treat each I_m independently. The triples $(x, y, z) \in \mathbb{Z}^3 - \{0\}$ such that $(x : y : z) \in \mathbb{P}^2$ has height $\leq N$ and is within $O(\delta)$ of $\phi(I_m)$ are among the nonzero integer points in a parallelepiped P_m of height, length and width proportional to $N, \delta^{1/2}N, \delta N$. Thus we expect that $|P_m \cap \mathbb{Z}^3|$ is approximately the volume of P_m , provided that this volume is $\gg 1$. This is the case once $\delta \gg N^{-2}$. (That is why we insisted that $\delta \gg N^{-2}$: choosing smaller δ would only make us work at least as hard to find fewer points.) Listing all the points in $P_m \cap \mathbb{Z}^3$ is a standard application of lattice reduction. Let M_m be an invertible 3×3 matrix such that $M_m P_m$ is the cube $K = [-1, 1]^3$. We are then seeking all $v \in \mathbb{Z}^3$ such that $M_m v \in K$, or equivalently all vectors in $K \cap M_m^{-1}\mathbb{Z}^3$. We find them by reducing the lattice $M_m^{-1}\mathbb{Z}^3$. This gives us a matrix $L_m \in \mathrm{GL}_3(\mathbb{Z})$ such that $M_m L_m$ is small. Now $M_m v \in K$ if and only if $w \in \mathbb{Z}^3 \cap (M_m L_m)^{-1}K$ where $v = L_m w$. But $(M_m L_m)^{-1}K$ is contained in the box centered on the origin whose i -th side is twice the l^1 norm of the i -th row of $(M_m L_m)^{-1}$ ($i = 1, 2, 3$). For each nonzero integral w in this box, calculate $(x, y, z) = L_m w$ and test whether $(x : y : z)$ in fact has height $\leq N$ and lies within δ of C . Doing this for each m yields the full list of such points.

As advertised, the algorithm requires only $O(\log N)$ space (though much more space is usually needed to store the results of the computation). Also, since each of many intervals I_m is treated independently, the computation can be massively parallelized with little loss among processors that interact only by reporting each $(x : y : z)$ to headquarters as it is found. How long do we expect the computation to take? We assume that ϕ and its derivatives can be calculated to within $N^{-O(1)}$ in time $\log^{O(1)} N$. Such is the case for all curves we consider and for every algebraic plane curve. Then each M_m takes only $\log^{O(1)} N$ operations to compute. Each lattice reduction can also be done in time polynomial in $\log N$, since our lattices are in fixed dimension — and moreover our dimension of 3 is small enough that Minkowski reduction is described explicitly. [For an overview and further references concerning Minkowski reduction, see [CS, pp.396–7].] So

¹ Our computations indicate that the first example is probably the smallest value of $|x^\pi + y^\pi - z^\pi|$ for positive integers x, y, z , and at any rate the smallest with $z \leq 10^6$; and the second is the smallest ratio of $|x^7 + y^7 - z^7|$ to z^7 , and even to z^4 , for positive integers satisfying $x \leq y < z \leq 10^6$. See the next section.

far this amounts to $\delta^{-1/2} \ll N$ time up to the usual log factors. Now each P_m has volume $2^3/|\det M_m| \ll (\delta^{1/2}N)^3$. If each $(M_m L_m)^{-1}$ had all of its entries $O(\delta^{1/2}N)$ — equivalently, if the shortest nonzero vectors of each lattice $M_m^{-1}\mathbb{Z}^3$ had length $\gg \delta^{-1/2}/N$ — then there would only be $O((\delta^{1/2}N)^3)$ choices for w , which summed over m gives $O(\delta N^3)$. Thus the total work would indeed be $\log^{O(1)} N$ times the expected number of solutions. Unfortunately it is too optimistic to expect that the entries of $(M_m L_m)^{-1}$ are all $\ll \delta^{1/2}N$. If the lattices $M_m^{-1}\mathbb{Z}^3$ are randomly distributed in the space of lattices of covolume $(\delta^{1/2}/N)^3$ in \mathbb{R}^3 , some of them will have nonzero vectors much shorter than $\delta^{-1/2}/N$. However, the *average* number of lattice vectors in K of a random lattice of determinant D is still $O(1/D)$. Thus we expect — and typically find in practice — that, even accounting for the occasional short lattice vector, we will find all rational points of height $\leq N$ that lie within δ of C , doing on average $\log^{O(1)} N$ work per point.

2.2 Can the Estimates Be Made Rigorous?

Our assumption that the lattices $M_m^{-1}\mathbb{Z}^3$ are randomly distributed was not proved; indeed it is false at least for some choices of C . Most glaringly, if C is a rational straight line then there are $\gg N^2$ rational points on C , and *a fortiori* at least as many at distance $\leq \delta$. While we of course will not apply our algorithm to straight lines, we do apply it to the n -th Fermat curve, which has contact of order n with several rational lines such as $y = z$; each of those lines contains $\gg N^{2-2/n}$ points at distance $\ll 1/N^2$ from the curve, exceeding the expected count of $N \log^{O(1)} N$ once $n > 2$. (These are the points we exclude by imposing the inequality $y < z$ in $0 < x \leq y < z < N$.) Assume, then, that C has at most finitely many tangent lines which have contact of order > 2 with C , and for any $\delta > 0$ let C_δ be the curve consisting of points of C at distance $\geq \delta$ from each of those higher-order tangent lines. For each point P on C_δ we obtain a lattice $L_\delta(P) \subset \mathbb{R}^3$ whose nonzero short vectors correspond to points near P in $\mathbb{P}^2(\mathbb{Q})$, of height $\ll \delta^{-1/2}$, lying at distance $\ll \delta$ from C_δ . This gives a map A_δ from C_δ to the moduli space of lattices in \mathbb{R}^3 . We would thus like to ask: as $\delta \rightarrow 0$, does the image of A_δ become uniformly distributed in this moduli space?

There are several problems with this formulation of our question. A minor one is that we have not defined A_δ precisely enough for the question to make sense, because we have left some O -constants unspecified. This did not matter for qualitative properties such as whether the lattice has $O(1)$ short vectors, but makes it easy to frustrate uniform distribution by simply choosing A_δ to avoid a small region in the moduli space. This problem is easy enough to fix for any given C ; for instance, if C is given by $x \mapsto (x : y(x) : 1)$ for some differentiable function $y : [0, 1] \rightarrow [-1, 1]$ with bounded second derivatives, we may take for $A_\delta(x)$ the integer span of the columns of

$$\begin{pmatrix} 0 & 0 & \delta \\ 1 & 0 & -x \\ -y'/\delta & 1/\delta & (xy' - y)/\delta \end{pmatrix}. \quad (1)$$

But this brings us to a more serious difficulty. The question of whether $\Lambda_\delta(C_\delta)$ is asymptotically uniformly distributed as $\delta \rightarrow 0$ is likely to be a very hard problem in analytic number theory. For our purposes we are only concerned with how often and how close does $\Lambda_\delta(P)$ come near the cusp of the moduli space. For instance, we see in the final section that if C is a conic then $\Lambda_\delta(C_\delta)$ is restricted to a surface in the moduli space of lattices in \mathbb{R}^3 , but within that surface it still approaches the cusp rarely enough that the average number of short vectors in a lattice in $\Lambda_\delta(C)$ is still $\ll \log(1/\delta)$. In general, then, what we would like is the following result: as $\delta \rightarrow 0$, the average number of vectors of norm < 1 of a lattice in $\Lambda_\delta(C_\delta)$ is $\ll \log^{O(1)}(1/\delta)$.

This still looks like a very difficult problem. While it remains open, we propose a contingency plan in case the lattices $L_\delta(P)$ have many more short vectors than expected. If all the short vectors are multiples of a single vector of small norm, there is no difficulty, because all these multiples yield the same point in \mathbb{P}^2 . But there could be two independent short vectors, whose linear combinations yield a line in \mathbb{P}^2 containing many points of small height near C . We claim that this is in fact the only way that a lattice of covolume $\ll 1$ could have more than $O(1)$ short vectors. This claim is easy enough to check using the description of Minkowski-reduced lattices in \mathbb{R}^3 , but we shall later need a generalization to lattices in higher dimension. We thus state and prove the generalization as follows:

Lemma 1. *For each positive integer n and positive real t there exists an effective constant $M_n(t)$ such that the following bound holds: for any lattice $\Lambda \subset \mathbb{R}^n$ whose dual lattice Λ^* has no nonzero vector of length $< r$, and for any $R > 0$, there are at most $M_n(rR) r^{-n} |\Lambda|^{-1}$ vectors of length $\leq R$ in Λ .*

Here $|\Lambda|$ is the covolume $\text{Vol}(\mathbb{R}^n/\Lambda)$. The lemma can be obtained as a consequence of the theory of lattice reduction, but it is not easy to extract $M_n(t)$ explicitly this way. We thus give the following alternative proof in the spirit of [C1] from which explicit (albeit far from optimal) bounds $M_n(t)$ may be easily computed if desired.

Proof. Given n , choose a positive Schwartz function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with the following properties: f is radial, i.e. $f(x)$ depends only on $|x|$; and the Fourier transform $\hat{f} : \mathbb{R}^n \rightarrow \mathbb{R}$, defined for $y \in \mathbb{R}^n$ by

$$\hat{f}(y) := \int_{x \in \mathbb{R}^n} f(x) e^{2\pi i (x, y)} dx, \quad (2)$$

satisfies $\hat{f}(y) \leq 0$ for all y such that $|y| \geq 1$. For instance, we may take

$$f(x) = (|x|^2 + a) e^{-\pi c |x|^2} \quad (3)$$

where

$$0 < c < \frac{2\pi}{n}, \quad a = \frac{1}{c^2} - \frac{n}{2\pi c}, \quad (4)$$

because the Fourier transform of a function (3) is

$$\hat{f}(y) = \left(a + \frac{n}{2\pi c} - \frac{|y|^2}{c^2} \right) e^{-\pi|y|^2/c} \quad (5)$$

for any $c > 0$ and $a \in \mathbb{R}$. By Poisson summation,

$$\sum_{x \in A} f(rx) = \frac{1}{r^n |\Lambda|} \sum_{y \in \Lambda^*} \hat{f}(y/r). \quad (6)$$

Under the hypothesis on r , the only positive term in the sum over y is $\hat{f}(0)$. The sum over x is bounded from below by the sum over x of length $\leq R$, which is at least the number of such vectors times $\min_{|x| \leq R} f(rx)$. It follows that Λ has at most

$$\frac{\hat{f}(0)}{r^n |\Lambda| \min_{|x| \leq rR} f(x)} = M_n(rR) r^{-n} |\Lambda|^{-1} \quad (7)$$

vectors of length $\leq R$, as claimed.

Corollary 1. *For each positive integer n there exists an effective constant A_n such that if a lattice $\Lambda \subset \mathbb{R}^n$ has more than $A_n R^n / |\Lambda|$ vectors of length $< R$ for some $R > 0$ then all those vectors lie in a hyperplane, which can be computed in polynomial time.*

Proof. Except for the last phrase, this follows from the previous Lemma by taking $r = 1/R$ and $A_n = M_n(1)$, since then Λ^* must have a nonzero y of length at most r , and any vector of Λ of length $< R$ must be orthogonal to y . To assure that y can be computed in polynomial time, we take $r = c/R$ for a positive constant c small enough that if Λ^* has a nonzero vector of length at most c/R then the LLL algorithm will find a (possibly different) nonzero vector of length at most $1/R$. Our Corollary now holds with $A_n = c^{-n} M_n(c)$.

From the case $n = 3$ of this Corollary we deduce:

Theorem 1. *Let C be the image of a differentiable map $\phi : [0, 1] \rightarrow \mathbb{RP}^2$ with bounded second derivatives. Then for each $N > 1$ and $\delta \geq N^{-2}$ one can find $O(\delta N^3)$ rational points and $O(N)$ rational line segments each of length $O(1/N)$ in \mathbb{P}^2 which together include all rational points of height $\leq N$ at distance $\leq \delta$ from C . These points and line segments can be computed in time $\ll \delta N^3 \log^{O(1)} N$. Outside of $O(\delta N^3 \log N)$ space used only to record each point or segment as it is found, the computation requires space $\ll \log^{O(1)} N$. All implied constants depend effectively on C .*

Note that here we do not exclude neighborhoods of high-order rational tangents to C ; such tangents will contain some of the lines segments computed by the algorithm.

2.3 Variations and Generalizations

The problem of finding rational points near plane curves is only the first non-trivial example of many analogous problems to which our method can apply. We briefly discuss some of these here.

One easy variation is to change the norm: instead of approximating the curve in the real valuation, use a nonarchimedean one, or a combination of several. For instance, one can efficiently seek nontrivial triples of small integers the sum of whose cubes is divisible by a high power of 2 or of 10. Likewise one can replace \mathbb{Z} by $\mathbb{F}_q[T]$ or similar rings in function fields of positive genus. The lattice-reduction step should then be even easier than in the archimedean case, though in the function-field setting our approach faces strong competition from the method of undetermined coefficients, and it is not clear which is superior. All these comments apply equally to the adaptation of our method to the problem of finding small nonzero values of $|x^3 - y^2|$, provided the characteristic is not 2 or 3. For the $|x^3 - y^2|$ problem, the work estimates are again rigorous; otherwise, they are still heuristic, but their analysis may be more tractable in the function-field case.

Higher dimensions present many new opportunities. The easiest generalization is to a \mathcal{C}^2 hypersurface in \mathbb{P}^{k-1} . Here we are seeking small values of a homogeneous function of k variables evaluated at an integral point. This time we chop the hypersurface into $O(\delta^{-(k-2)/2})$ chunks each of diameter $O(\delta^{1/2})$, and replace each chunk by a subset of a hyperplane which approximates it to within $O(\delta)$. The points of height $\leq N$ that are within $O(\delta)$ of this chunk then come from integral points in a parallelepiped in \mathbb{R}^k whose sides have lengths $O(N)$, $O(N\delta^{1/2})$ ($k-2$ times), and $O(N\delta)$. Again most of these this parallelepipeds have volume $\gg 1$ provided $\delta \gg N^{-2}$, and we locate the integral points using lattice reduction in \mathbb{R}^k . So, as long as $\delta \gg N^{-2}$, we expect to find on the order of δN^k points, using $\ll \log^{O_k(1)} N$ space and spending $\ll \log^{O_k(1)} N$ time per point. For a general hypersurface, this again improves on other approaches to the problem. But the improvement decreases with k : the direct approach takes time $N^{k-1} \log^{O(1)} N$, and we lower the exponent by a factor no better than $(k-2)/(k-1)$, which approaches 1 as $k \rightarrow \infty$. Moreover, lattice reduction in \mathbb{R}^k quickly becomes difficult as k grows. Another consideration is that for special surfaces there are known, and simpler, algorithms that take time $N^{k-1} \log^{O(1)} N$ or less once $k > 3$. For instance, for Fermat surfaces in \mathbb{P}^3 , one readily adapts the method of [B1] to find all solutions of $x^n + y^n = z^n \pm t^n + O(z^{n-2})$ in positive integers with $t \leq z \leq N$, in expected time $N^2 \log^{O(1)} N$, and with no need for lattice reduction in \mathbb{R}^4 or other complicated ingredients. This computation does require space proportional to $N \log N$, which however poses no difficulty for practical values of N . As in the previous paragraph, all that is described in the present paragraph can be done also for a nonarchimedean norm, with similar results except that lattice reduction over a function field is tractable even for large k . In either case rigorous estimates may become even less accessible as k grows.

We can generalize further to manifolds $\mathcal{M} \subset \mathbb{P}^{k-1}$ of codimension $c > 1$. Here we expect to find on the order of $\delta^c N^k$ rational points of height $\leq N$ at distance $O(\delta)$ from \mathcal{M} . We chop \mathcal{M} into $O(\delta^{-(k-1-c)/2})$ patches of diameter $O(\delta^{1/2})$, each of which yields a parallelepiped in \mathbb{R}^k with dimensions of order N (once), $N\delta$ (d times), and $N\delta^{1/2}$ (the remaining $k-1-c$ dimensions). We thus expect to efficiently find all $\sim \delta^c N^k$ points as long as $\delta \gg N^{-2k/(k+c-1)}$. A further possibility emerges if \mathcal{M} has bounded derivatives past the second derivatives and has small enough dimension compared with k : we can then make further headway when δ falls below that threshold. Usually we are only interested in points much closer than $N^{-2k/(k+c-1)}$; but as long as we use only the \mathcal{C}^2 structure we gain nothing by making δ even smaller, so we may as well find all the points at distance $O(N^{-2k/(k+c-1)})$ and locate the best approximations in the resulting list. However, if \mathcal{M} is \mathcal{C}^3 and its dimension $d = k-1-c$ is so small that $k > \binom{d+2}{2}$, then a patch of diameter ϵ is contained in a box with d sides of length $\ll \epsilon$, a further $(d^2+d)/2$ sides of length $\ll \epsilon^2$, and the remaining $k - \binom{d+2}{2}$ sides of length $\ll \epsilon^3$. This means that we can make our parallelepipeds thinner in some directions, and thus use wider patches of \mathcal{M} , covering the entire manifold with fewer of them. This lets us locate the points of height $\leq N$ closest to \mathcal{M} in time significantly less than it would take to record all the points at distance $\ll N^{-2k/(k+c-1)}$, even though not so efficiently that we only spend $\ll \log^{O_k(1)} N$ time per point. More generally if \mathcal{M} is a \mathcal{C}^i manifold we can exploit bounds on the i -th derivatives once $k > \binom{d+i-1}{i-1}$.

If the ambient projective space is not of high enough dimension, we can still make some use of approximations to \mathcal{M} of degree $i > 1$ by using the i -th Veronese embedding V_i of \mathbb{P}^{k-1} into projective space of dimension $\binom{k+i-1}{i} - 1$. [The i -th Veronese embedding takes the point with projective coordinates $(X_1 : \dots : X_k)$ to the point whose projective coordinates are all $\binom{k+i-1}{i}$ monomials of degree i in the X_j . Thus V_i raises all heights to the power i , and transforms intersections with hypersurfaces of degree d in \mathbb{P}^{k-1} into hyperplane sections in a projective space of much higher dimension. For more on Veronese embeddings, see for instance [FH], where they arise several times.] The idea is to surround each patch of $V_i(\mathcal{M})$ by a box containing all points in $V_i(\mathbb{P}^{k-1})$ at distance $O(\delta)$ from $V_i(\mathcal{M})$. The resulting asymptotic improvement may be only barely worth it in practice, though. Consider the simplest case of a \mathcal{C}^3 curve $C \in \mathbb{P}^2$, embedded in \mathbb{P}^5 by V_2 . Assume for simplicity that the parametrization ϕ of C has $|\phi''|$ bounded away from zero. Then, for δ such that $\epsilon^3 \ll \delta \ll \epsilon^2$, the radius- δ neighborhood in \mathbb{P}^2 of an interval of length ϵ on C maps into a box in \mathbb{P}^5 whose sides are of order $\epsilon, \epsilon^2, \delta, \epsilon^3, \epsilon^4$. [To see this, choose coordinates $(X_0 : X_1 : X_2)$ on \mathbb{P}^2 for which ϕ is of the form $(1 : t : t^2 + O(t^3))$ for t in a neighborhood of 0, and note that V_2 takes $(X_0 : X_1 : X_2)$ to $(X_0^2 : X_0 X_1 : X_0 X_2 : X_1^2 : X_1 X_2 : X_2^2)$.] Thus the points of height at most N in that neighborhood map to lattice points in a 6-dimensional parallelepiped of volume $\ll \delta N^{12} \epsilon^{10}$. (Here N occurs to the power 12 rather than 6 because V_2 squares the height of each rational point.) Thus if we take $\epsilon = (\delta N^{12})^{-1/10}$ we expect to find all points at distance $\ll \delta$ from C , of which there should be about δN^3 , in time $N(\delta N^2)^{1/10} \log^{O(1)} N$. The

condition $\delta \gg \epsilon^3$ yields $\delta \gg N^{-36/13}$, so we save a factor of at most $N^{1/13}$. We pay not only by missing the points at distance between $N^{-36/13}$ and N^{-2} (which usually do not interest us anyway) but also by reducing lattices of rank 6 rather than 3. This takes more time per lattice, and probably yields parallelepipeds whose average bounding box is larger. Each of these effects amounts to only a constant factor, but these factors may be considerable, and it will be interesting to see how large N must be for this use of V_2 to be practical.

2.4 Rational Points on Varieties

In the last paragraph we exploited the fact that points near \mathcal{M} map under V_i to points that are not only near $V_i(\mathcal{M})$ but exactly on $V_i(\mathbb{P}^{k-1})$. We can go much further when we search for points exactly on \mathcal{M} . Again we consider the simplest case of a curve. We begin with a curve in one projective space:

Theorem 2. *Let C be an algebraic curve in M -dimensional projective space, defined over \mathbb{Q} and not contained in any hyperplane. Then for any $N \geq 1$ the rational points of C of height at most N can be listed in time $\ll_C N^{2/M} \log^{O_M(1)} N$. The implied constants depend effectively on d and C .*

Remarks. As seen above for $M = 2$, this result applies more generally to a \mathcal{C}^M curve in \mathbb{P}^M whose intersection with any hyperplane can be computed in polynomial time. The exponent $2/M$ is best possible: a rational normal curve of degree M (a.k.a. the image of \mathbb{P}^1 under V_M) has on the order of $N^{2/M}$ rational points of height at most N , and it takes time $\gg N^{2/d} \log N$ just to write them down. The constant implied in $O_d(1)$ and/or \ll_C , while effective, may be unpleasant in practice for large M , since lattice reduction in dimension $M + 1$ is involved.

Proof. A segment of C of length $\ll N^{-2/M}$ is contained in a box whose i -th side is $\ll N^{-2i/M}$ ($i = 1, 2, \dots, M$). The rational points of height at most N in this box come from points of \mathbb{Z}^{M+1} contained in a box B whose i -th side is $\ll N^{1-2i/M}$ ($i = 0, 1, 2, \dots, M$) and thus has volume $O(1)$. It takes time $\ll_C \log^{O_M(1)} N$ to apply lattice reduction and, by Corollary 1, either list $\mathbb{Z}^{M+1} \cap B$ or find a hyperplane containing $\mathbb{Z}^{M+1} \cap B$. In the former case, we test whether each of the resulting $O_d(1)$ points lies in C . In the latter case, we map this hyperplane to \mathbb{P}^M and intersect it with C , finding at most $\deg(C) = O_C(1)$ rational points. Thus in either case we find all rational points of height $\leq N$ on our segment in time $\ll_C \log^{O_M(1)} N$. Since it takes only $O(N^{2/M})$ segments to cover C , we are done.

It might seem that this algorithm is superfluous: if C has genus 0 then its small rational points may be found directly from a rational parametrization, without any lattice reduction; and if C has positive genus then we can find all its points of height $\leq N$ in time $\ll \log^{O(1)} N$ once we have generators of the Mordell-Weil group of the Jacobian of C . But the difficulty is that we must first find these generators, and this requires locating rational points on a curve

or a higher-dimensional variety. For instance, to find the Mordell-Weil group of an elliptic curve E we usually apply a few descents and then search for points on certain principal homogeneous spaces for E , each of which is a curve C of genus 1, usually (in the case of a complete 2-descent) of the form $y^2 = P(x)$ for some irreducible quartic $P \in \mathbb{Z}[X]$. One then searches for $x \in \mathbb{Q}$ of height up to H for which $P(x) \in \mathbb{Q}^2$. There are on the order of H^2 candidates for x ; one can set up a sieve to efficiently try them all, but this still takes time $H^2 \log^{O(1)} H$ (and significant space). Instead we can embed C in \mathbb{P}^3 as the intersection of two quadrics (by writing $P(x)$ as a homogeneous quadric in $1, x, x^2$), and use the algorithm of Thm.2 with $N = H^2$ to find all rational solutions of $y^2 = P(x)$ with x of height $\leq H$ in time $H^{4/3} \log^{O(1)} H$. For certain E one can use Heegner points to locate a rational point on C to within δ (see [E3]); if $\delta \ll H^{-2}$, this is sufficient to identify x using continued fractions, a.k.a. lattice reduction in dimension 2. Using the new algorithm, we see that $\delta \ll H^{-4/3}$ suffices if we use lattice reduction in dimension 4. This saves a constant factor in the computation of x , since fewer digits and terms are needed in the floating-point computation of Heegner points. When C has genus > 1 , there are only finitely many rational points by Faltings' theorem, but they still may be of significant number and/or height. For instance, in [KK,S] one finds curves $C : y^2 = P(x)$ of genus 2 which have hundreds of rational points. In both cases, all points with x of height $\leq 10^6$ were found using the $H^2 \log^{O(1)} H$ sieve method, a substantial computation. At least in the case considered in [S], where the Jacobian of C is absolutely simple with large Mordell-Weil rank, it would probably be even more onerous to find all these points by first determining the Mordell-Weil group. But the embedding $(1 : x : x^2 : x^3 : y)$ of C into \mathbb{P}^4 yields an improvement from H^2 to $H^{3/2}$ with 5-dimensional lattice reduction.

We can do even better by mapping the same curve to larger projective spaces. Fix an algebraic curve C of genus g defined over \mathbb{Q} , and a divisor D on C of degree $d > 0$. For n sufficiently large, the sections of nD embed C into \mathbb{P}^{nd-g} . This embedding sends any rational point on C of height (exponential, as usual here) $\leq H$ relative to D to a point on \mathbb{P}^{nd-g} of height $\ll H^n$. By Thm.2 again, we can find all such points in time $\ll H^{2n/(nd-g)} \log^{O(1)} H$. Letting $n \rightarrow \infty$, we conclude:

Theorem 3. *Fix an algebraic curve C/\mathbb{Q} and a divisor D on C of degree $d > 0$. For each $\epsilon > 0$ there exists an effectively computable constant A_ϵ such that for any $H \geq 1$ one can find all points of C whose height relative to D is at most H in time $A_\epsilon H^{(2/d)+\epsilon}$.*

For instance, all rational points on $y^2 = P(x)$ with x of height at most H can be computed in time $\ll_\epsilon H^{1+\epsilon}$.

What of varieties \mathcal{M} of dimension $\Delta > 1$ in \mathbb{P}^M ? A chunk of radius δ then yields the intersection of \mathbb{Z}^{M+1} with a box with sides as follows: one of length $O(N)$, Δ sides of length $O(N\delta)$, $\binom{\Delta+1}{2}$ sides of length $O(N\delta^2)$, \dots , $\binom{\Delta+i-1}{i}$ sides of length $O(N\delta^i)$, \dots until $\binom{\Delta+j}{j} = \sum_{j=0}^i \binom{\Delta+i-1}{i}$ first exceeds M . As usual we choose δ so that the product of these sides is 1, and apply lattice

reduction to each of $O(\delta^{-\Delta})$ chunks. The difficulty here is that if the lattice is nearly degenerate, the hyperplane found in Corollary 1 meets \mathcal{M} not in a finite number of points but in a subvariety of positive dimension $\Delta - 1$. This suggests an induction on Δ , since we can apply our method to that hyperplane section of \mathcal{M} . But already for $\Delta = 2$ such an argument requires a version of Thm.2 with more uniformity in the implied constants than we know how to obtain. However, as with our first nontrivial case of curves in \mathbb{P}^2 , we do not expect such degenerate lattices to arise in practice often enough to raise the computational cost above $O(\delta^{-\Delta} \log^{O(1)} N)$, except for a finite number of proper subvarieties of \mathcal{M} . If we assume this, we can again obtain better estimates by embedding \mathcal{M} in larger projective spaces. Fix an ample divisor D on \mathcal{M} , and ask for all rational points whose height relative to \mathcal{M} is at most H . Using the sections of nD to embed \mathcal{M} in projective spaces, and letting $n \rightarrow \infty$, we find the following heuristic generalization of Thm.3: for each $\epsilon > 0$, there exists a proper subvariety $\mathcal{M}_0(\epsilon)$ of \mathcal{M} such that all points of $\mathcal{M} - \mathcal{M}_0(\epsilon)$ of height at most H relative to D can be found in time

$$O_\epsilon(H^{((\Delta+1)/|D|)+\epsilon}), \quad (8)$$

where $|D|$ is the Δ -th root of the intersection number D^Δ . One might even hope that $\mathcal{M}_0(\epsilon)$ can be taken independent of ϵ . For instance, if \mathcal{M} is a surface of degree d in \mathbb{P}^3 then we expect that, for some union \mathcal{M}_0 of curves on \mathcal{M} , we can find all rational points of height $\leq N$ on $\mathcal{M} - \mathcal{M}_0$ in time $\ll_\epsilon N^{(3/\sqrt{d})+\epsilon}$. We must admit that this is unlikely to yield a practical improvement over the $N^2 \log^{O(1)} N$ method we already knew: the first V_i that reduces the exponent of N below 2 is V_3 , and then (assuming $d \geq 4$) the exponent drops only to $24/13$ — but instead of reducing 4-dimensional lattices we are then faced with lattice reduction in dimension 20. It will probably be a long time before N can feasibly be taken large enough that this extra effort is worth the $N^{2/13}$ factor gained.

Returning to plane curves, we can use this idea to prove an even stronger bound on rational points on a plane curve C that is analytic but not algebraic. This is because the homogeneous monomials of degree i in the coordinates of C are linearly independent for each i , so $V_i(C)$ spans a projective space whose dimension grows quadratically in i (whereas for an algebraic curve the growth is always linear). This leads us to the following result:

Theorem 4. *Let C be a transcendental analytic arc in \mathbb{P}^2 , i.e. $C = \{f(x) : a \leq x \leq b\}$ where f is an analytic map from a neighborhood of $[a, b]$ to \mathbb{P}^2 whose image is contained in no algebraic curve. Then for each $\epsilon > 0$ there exists a constant A_ϵ such that for every $H \leq 1$ there are fewer than $A_\epsilon H^\epsilon$ points of height $\leq H$ in $C \cap \mathbb{P}^2(\mathbb{Q})$.*

Proof. For each positive integer i consider $V_i(C) \subset \mathbb{P}^{(i^2+3i)/2}$. Since C is transcendental, $V_i(C)$ is an analytic arc $V_i \circ f$ contained in no hyperplane of $\mathbb{P}^{(i^2+3i)/2}$. Now apply the argument for Thm.2 with $N = H^i$. As noted in the remarks following the statement of that theorem, the curve need not be algebraic as long as it is \mathcal{C}^M and its intersection with any hyperplane is of bounded size. (Here

we need not compute this intersection numerically, since we are only bounding the number of rational points of small height on C , not computing them efficiently.) The differentiability is clear since $V_i(C)$ is analytic, and the boundedness is proved in the next lemma. We conclude that the number of points of height $\leq H$ on C is $\ll_i H^{4/(i+3)}$. Since i can be taken arbitrarily large, our theorem follows.

The existence of an upper bound on the size of the intersection of any hyperplane with $V_i(C)$ is a special case of the following lemma in complex analysis. Throughout the lemma and its proof we count zeros of an analytic function according to multiplicity, even though in the application to Thm.4 a multiple zero is no worse than a simple one.

Lemma 2. *Let E be an open subset of \mathbb{C} and V a finite-dimensional vector space of analytic functions: $E \rightarrow \mathbb{C}$. Then for any compact subset $K \subset E$ there exists an integer n such that any nonzero $f \in V$ has at most n zeros in K .*

Proof. Fix K . We shall say that a compact $K' \subset E$ is “good” if its boundary $\partial(K')$ is rectifiable and its interior $K' - \partial(K')$ contains K . Choose a good K_1 , and define a norm on V by $\|f\| = \sup_{z \in K'} |f(z)|$. Let V_1 be the unit ball $\{f \in V : \|f\| = 1\}$. It is sufficient to prove the lemma for $f \in V_1$.

For each $f \in V_1$ choose a good $K_f \subseteq K'$ such that f does not vanish on $\partial(K_f)$. Let $r_f = \inf_{z \in \partial(K_f)} |f(z)|$, and let n_f be the number of zeros of f in K_f . By Rouché’s theorem, if $g \in V$ with $\|f - g\| < r_f$ then g has at most n_f zeros in K_f , and *a fortiori* in K . Now V_1 is compact and is covered by the open balls B_f of radius r_f about $f \in V_1$. Thus there is a finite subcover $\{B_{f_i}\}_{i=1}^M$. Then $n := \max_i n_{f_i}$ is an upper bound for the number of zeros in K of any $f \in V_1$, and thus of any nonzero $F \in V$.

To recover our result on hyperplane sections of $V_i(C)$, take $K = [a, b]$, let E be a neighborhood of K on which f is analytic, and choose any analytic functions f_0, f_1, f_2 on E such that $f = (f_0 : f_1 : f_2)$ on E . Then take for V the space of homogeneous polynomials of degree i in f_0, f_1, f_2 . If we understand f well enough to obtain for each i an effective bound n in Lemma 2 then the constants A_ϵ in Thm.4 are effective too.

With a little additional work \mathbb{Q} can be replaced by an arbitrary number field F embedded in \mathbb{C} , and C by $f(K)$, where $K \subset \mathbb{C}$ is any compact subset and f is again an analytic map from a neighborhood of K to \mathbb{P}^2 whose image is contained in no algebraic curve.

A separate approach to bounding the number of rational points on curves was initiated in [BP] and pursued further in [P] and [HB2]. For example, Heath-Brown obtains in [HB2] bounds on the number of rational points on an algebraic plane curve that coincide with the time estimates in our Theorems 2 and 3. Moreover, our Thm.4 is contained in [P, Thm.8], which asserts that for a number field F with $[F : \mathbb{Q}] = n$ the number of F -rational points of height $< H$ on a transcendental analytic arc C is at most $A_{C,n,\epsilon} H^\epsilon$. Probably the methods

of [BP,P] can also prove these results with arcs C replaced by compact transcendental curves $f(K)$, and our bounds can also be made uniform in F given $[F : \mathbb{Q}]$. There is clearly some overlap between the two approaches; for instance the Corollary preceding [P, Thm.8] is the same as our Lemma 2, but proved using the determinants of [BP,P]. What is not clear, but intriguing, is whether those determinantal methods and our lattice-reduction technique can ultimately be interpreted as facets of the same basic idea.

All this also suggests the question of whether a transcendental arc can contain infinitely many rational points, of whatever height. I thank Michel Waldschmidt for pointing out that this question was already asked, and later answered affirmatively, by Weierstrass. See [M2, Chapter 3] for this and related results.

3 The Algorithm in Practice

In this section we report on the outcome of the application of our algorithm to various plane curves, and on some results suggested by our findings. We suppress details of the explicit constants replacing each $O(\dots)$ and \ll ; these details are of course crucial in practice, but are straightforward and not enlightening. In each case our curve has some rational points of inflection, and we make sure to truncate our curve enough to avoid the tangents at such points but not so much that we lose approximations near but not on those tangents.

In general, for a plane curve given by a homogeneous equation $P(X, Y, Z) = 0$ of degree n , we associate to a rational point $(x : y : z)$ near but not on the curve the number

$$n \max(|x|, |y|, |z|)^{n-3} / |P(x, y, z)|, \quad (9)$$

which measures how close the point $(x : y : z)$ is to the curve relative to the point's height. We insert the factor n so that we can reasonably compare approximations for curves of different degrees. For instance, for the Fermat curve one expects that as x, y vary, the integer $z^n := x^n + y^n$ comes on average within $\frac{1}{4}nz^{n-1}$ of the nearest n -th power of an integer, and thus that the smallest value of $|z^n - y^n - x^n|$ for $z \in [N, 2N]$ is proportional to nz^{n-3} . One could insert further factors to correct for the length and shape of our curve, but these factors are not significant for most of the curves we study.

We noted already that the heuristics leading to formulas such as (9) refer to “random” $(x : y : z)$ near the curve, not for systematic families of approximations which may attain values of the ratio (9) larger or more often than expected. We again give an example for the Fermat curves, which were the subjects of most of our computations. One usually guesses that for each r there will be $\ll r \log N$ triples (x, y, z) such that the ratio (9) exceeds r . However, in the identity

$$(t+1)^n - (t-1)^n = 2nt^{n-1} + O(t^{n-3}), \quad (10)$$

we can make $2nt^{n-1}$ an n -th power by setting $t = 2nu^n$; this yields $\gg N^{1/n}$ triples with (9) bounded away from zero. We note the special cases $n = 2, 3$ of this identity: for $n = 2$, the $O(t^{n-3})$ error vanishes, and we recover a familiar

parametrization of Pythagorean triples; for $n = 3$, the error is constant, and we can scale the identity to obtain the known family of solutions $(x, y, z) = (6t^2, 6t^3 - 1, 6t^3 + 1)$ of $z^3 - y^3 - x^3 = 2$. Returning to general n : in our searches we set the threshold on $z^{n-3}/|z^n - y^n - x^n|$ low enough to find all the examples coming from (10), as a check on the computation; but we chose a higher threshold for the tabulation of results so that our list is not dominated by this polynomial family.

3.1 Fermat Curves of Degree > 3

We implemented our algorithm to find small values of $|z^n - y^n - x^n|$ with $0 < x \leq y < z$, $4 \leq n \leq 20$, and $z \in [10^3, 10^6]$. Since the threshold for “small” depends on the size of z , we wrote $[10^3, 10^6]$ as the union of 10 intervals $[N/2, N]$ and treated each separately. We also used a direct search for $z < 5000$, using the overlap region $[1000, 5000]$ as a check on the computation. We did not attempt to fine-tune the algorithm for efficiency, since we carried it out more as a demonstration project than a major computational undertaking. Thus we programmed the search in **gp**, using the built-in arithmetic and LLL lattice reduction. We estimate that transcribing the program to C, and replacing LLL by Minkowski reduction in \mathbb{R}^3 , would speed the computation by roughly an order of magnitude; of course a machine faster than a Sun Sparcstation Ultra 1 would help too. With a C program and a more powerful machine, it should be feasible to search the range $n \in [4, 20]$, $z < 10^9$ in time on the order of a month.

The behavior of the run times and the counts of solutions with $|z^n - y^n - x^n| \ll z^{n-2}$ seem broadly consistent with our heuristics, though we have not attempted a detailed statistical analysis. We tabulate the most striking examples, those with

$$r := nz^{n-3}/(z^n - y^n - x^n) \tag{11}$$

of absolute value at least 4:

All decimal values of r are rounded to the nearest tenth. If for some integer $\lambda > 1$ we have $r > 4\lambda^3$ then $(\lambda x, \lambda y, \lambda z)$ will also appear in the table provided $\lambda z \leq 10^6$; this happens for $\lambda = 2$ at $n = 4, 5, 7, 10$, and for $\lambda = 3$ at $n = 5, 10$. The first examples for $n = 10$ and particularly $n = 5$ (where $13^5 + 16^5 = 17^5 + 12$) are small and striking enough that one feels they must have been observed already, but I do not know a reference. On the other hand, the first two examples for $n = 12$ have been published, and in a most unlikely place: each appeared in a different episode of the popular animated cartoon *The Simpsons*. Perhaps the third example for $n = 12$, or an example with $n = 7$ or $n = 15$, could be used if the cartoon repeats this theme once more; the relative error $|z^n - y^n - x^n|/z^n$ in each case is between 1 and 2 parts in 10^{18} , as compared to $3 \cdot 10^{-10}$ and $2 \cdot 10^{-11}$ for the two four-digit examples...

n	x	y	z	r	n	x	y	z	r
4	167	192	215	-4.5	8	209959	629874	629886	-11.6
4	8191	16253	16509	12.9	8	209945	629826	629838	11.6
4	24576	48767	49535	-64.5	9	6817	10727	10747	5.3
4	49152	97534	99070	-8.1	9	21860	25208	25903	24.7
4	34231	157972	158059	5.2	10	280	305	316	137.1
4	76215	311390	311669	-14.8	10	560	610	632	17.1
5	13	16	17	-120.4	10	840	915	948	5.1
5	26	32	34	-15.1	10	7533	8834	8999	4.4
5	39	48	51	-4.5	12	1782	1841	1922	6.1
5	42	71	72	-8.8	12	3987	4365	4472	-7.1
5	262	328	347	-6.2	12	781769	852723	874456	10.3
5	1125	2335	2347	-5.0	13	666	806	811	8.3
5	5088	16155	16165	4.1	13	5579	8235	8239	4.1
5	190512	292329	298900	5.5	15	434437	588129	588544	42.9
6	1236	3587	3588	12.5	16	492151	741267	741333	4.6
6	6107	8919	9066	-9.9	19	79	85	86	-4.7
7	386692	411413	441849	78.4	19	491	565	567	4.9
7	773384	822826	883698	9.8	19	43329	51144	51257	5.8
					20	4110	4693	4709	4.3

Frivolity aside, one is struck by the pair of examples for $n = 8$. The values of r are far from the largest in the table, but they are almost equal and opposite, and involve nearly equal triples (x, y, z) for which $z - y$ has the same small value of 12. This suggests that we are dealing with a polynomial family $(x(t), y(t), z(t))$ specialized at $t = \pm t_0$. Indeed we quickly find that these are the cases $t = \pm 3$ of

$$(32t^9 + 6t)^8 + (32t^8 + 7)^8 = (32t^9 + 10t)^8 + 21 \cdot 2^{28}t^{40} + O(t^{32}), \quad (12)$$

with $r = t^5/21 + O(t^{-3})$. Thus arbitrarily large values of r occur, and indeed $z^8 - y^8 - x^8$ can be as small as $O(z^{40/9})$ rather than the expected $O(z^5)$. Trying to generalize the identity (12) further, we soon find that there are similar families for any exponent n such that $3n(n-2)$ is a square:

Theorem 5. *Let $n > 1$ be a positive integer. Then there exist polynomials $x(t), y(t), z(t) \in \mathbb{Z}[t]$ of the form*

$$x(t) = Ct^n + D, \quad y(t) = At^{n+1} + Bt, \quad z(t) = At^{n+1} + B't \quad (13)$$

with $A \neq 0, B' \neq B$ such that $z^n - y^n - x^n$ is a polynomial of degree at most $n(n-3)$, if and only if $3n(n-2)$ is a square. In that case, there exist infinitely many integer triples (x, y, z) with $0 < x < y < z$ such that $z^n - y^n - x^n \ll z^{(n^2-3n)/(n+1)}$.

Proof. Let b, b' be the distinct rational numbers $B/A, B'/A$. Expand $z^n - y^n$ at infinity:

$$\begin{aligned} z^n - y^n &= nA^n(b' - b) \left(t^{n^2} + \frac{n-1}{2}(b' + b)t^{n^2-n} \right. \\ &\quad \left. + \frac{(n-1)(n-2)}{6}(b'^2 + b'b + b^2)t^{n^2-2n} + O(t^{n^2-3n}) \right). \end{aligned} \quad (14)$$

For this to be of the form $(Ct^n + D)^n + O(t^{n^2-3n})$ we must have

$$(n-1) \left(\frac{n-1}{2} (b' + b) \right)^2 = 2n \frac{(n-1)(n-2)}{6} (b'^2 + b'b + b^2). \quad (15)$$

The discriminant of this quadratic equation in b'/b is $3n(n-2)$ times a square; thus (15) has nonzero rational solutions if and only if $3n(n-2)$ is a square. Explicitly we find that b, b' are proportional to $\sqrt{(n^2 - 2n)/3} \pm 1$.

Conversely, suppose $n^2 - 2n = 3m^2$ for some integer m . Let

$$z = A(t^{n+1} + c(m+1)t), \quad y = A(t^{n+1} + c(m-1)t). \quad (16)$$

Then

$$z^n - y^n = 2cnA^n \left(t^n + \frac{n-1}{n} cm \right)^n + O(t^{n^2-3n}). \quad (17)$$

To make this $(Ct^n + D)^n + O(t^{n^2-3n})$ with $C, D \in \mathbb{Z}$ we now need only choose nonzero $c \in \mathbb{Z}$ so that $2cn$ is an n -th power (e.g. take $c = (2n)^{n-1}$), and then choose A so that $n|Acm$. Specializing t to sufficiently large integers in the resulting $(x(t), y(t), z(t))$ yields infinitely many integer triples (x, y, z) with $0 < x < y < z$ such that $z^n - y^n - x^n \ll z^{(n^2-3n)/(n+1)}$, as claimed. \square

The smallest $n > 3$ such that $n^2 - 2n = 3m^2$ is $n = 8$. There are infinitely many further examples, starting with 27, 98, 363, ..., and parametrized by a Fermat-Pell equation. Dropping the constraint $n > 3$ yields the further cases $n = 2$ and $n = 3$. For $n = 2$ we again obtain a Pythagorean parametrization, this time with x, y, z multiplied by t ; for $n = 3$ we find

$$(9t^3 + 1)^3 + (9t^4)^3 - (9t^4 + 3t)^3 = 1, \quad (18)$$

one of infinitely many polynomial solutions of $x^3 + y^3 - z^3 = 1$.

3.2 The Fermat Cubic

Our algorithm applies to the Fermat cubic as it does to the Fermat curves of higher degree, but we treat it separately both because the heuristic analysis is subtler and because the problem of finding small values of $|z^3 - y^3 - x^3|$ has already attracted some attention. We noted that in general we expect the smallest values of $|z^n - y^n - x^n|$ to be comparable with z^{n-3} . For $n = 3$, we have $z^{n-3} = 1$, and of course (given this case of Fermat's Last Theorem) $|z^3 - y^3 - x^3|$ can be no smaller than 1 for nonzero integers x, y, z . Moreover, $z^3 - y^3 - x^3$ cannot be an arbitrary rational multiple of z^{n-3} : only the discrete values $\pm 1, \pm 2, \dots$ may arise. Thus, instead of a Diophantine inequality $z^n - y^n - x^n \ll z^{n-3}$, we have a family of Diophantine equations $z^3 - y^3 - x^3 = d$ ($d \in \mathbb{Z}$), and new tools can bear on solving them or, failing that, describing their distribution of solutions. These equations have been investigated by various means since the beginning of the computer age; see [G] for references to work up to about 1980 (some of which dates back to the 1950's), and [B2,CV,HBLR,KTS,PV] for more recent results.

As we shall see, the problem has been approached in several ways, some of which already improve on direct exhaustion over some N^2 values of (x, y) . Still, our new linear approximation method is better yet, both in heuristic theory — even though by factors smaller than our accustomed $N/\log^{O(1)} N$ — and in practice, as evidenced by the computation of many new solutions. Our discussion here applies with almost no change to other “diagonal” cubics, such as $x^3 + y^3 + 2z^3$ which was also singled out in [G, Prob. D5]; but we have not yet implemented a search for small values of $|x^3 + y^3 + 2z^3|$ beyond what has already been reported in the literature.

For each nonzero d , the expected distribution of solutions of

$$z^3 - y^3 - x^3 = d \quad (19)$$

involves not only considerations of size — i.e. of local behavior at the archimedean place of \mathbb{Q} — but also on the behavior of $z^3 - y^3 - x^3$ at finite primes p : each p contributes a local factor $f_p(d)$ that is the ratio of the p -adic measure of the \mathbb{Z}_p -points of (19) to the average of that measure as d ranges over \mathbb{Z}_p . For instance, if any of those factors $f_p(d)$ vanishes, there can be no solutions at all. It is not hard to see that the only such local constraint is $d \not\equiv \pm 4 \pmod{9}$. For such d , the resulting product over p was investigated by Heath-Brown [HB1]. He showed that the product does not converge absolutely, but can nevertheless be analyzed and approximated numerically by comparing $f_p(d)$ with the factor at p of the Euler product for $(\zeta_{\mathbb{Q}(\sqrt[3]{d})}(s)/\zeta(s))^3$ at $s = 1$, which differs from $f_p(d)$ by a factor of at most $1 + O(p^{-3/2})$. The product $\prod_p f_p(d)$ is then seen to diverge to $+\infty$ if d is a cube and to converge to a positive limit when d is neither a cube nor congruent to $\pm 4 \pmod{9}$. Heath-Brown thus conjectured in [HB1] that all nonzero integers $d \not\equiv \pm 4 \pmod{9}$ occur as $z^3 - y^3 - x^3$ infinitely often. So far this is only known when d is either a cube or twice a cube, thanks to polynomial parametrizations, which the above heuristics do not try to account for. We have already exhibited polynomial solutions for $d = 1, 2$. For many $d \not\equiv 4 \pmod{9}$ which are neither cubes nor twice cubes, not a single solution is known for $z^3 - y^3 - x^3 = d$. Heath-Brown observes [HB1] that this is not surprising, because for many of these d the expected number of solutions with $z \in [N, 10^6 N]$ is positive but smaller than 1. Guy [G] lists the cases with $d < 10^3$ which were open as of 1980, and while the list is now shorter the question of which integers are the sums of three cubes is not yet settled even in that range. For instance, the case $d = 30$ was open until 1999, and had been the smallest open case for several decades.

We have noted already that a direct search finds all small $|z^3 - x^3 - y^3|$ with $z < N$ in time $N^2 \log^{O(1)} N$. There have been several improvements on this, all obtained by rewriting the equation (19) as

$$x^3 + d = z^3 - y^3 = (z - y)(z^2 + yz + y^2). \quad (20)$$

Once $x^3 + d$ is factored, which takes heuristic time $N^{o(1)}$, all solutions of (20) can be found by trying each factor of $x^3 + d$ for $z - y$. Given the value of d , this takes time only $N^{1+o(1)}$. In addition to dealing with only one d at a time, this

method has the disadvantage that the $N^{o(1)}$ time required to factor x^3+d , though subexponential, is still considerable. The advantage of this method is that it finds all solutions with $x \leq N$, while y, z may be considerably larger, of order up to $N^{3/2}$. Many of the new solutions found in [KTS] are of this type, with y, z large but $z-y$ very small. Heath-Brown observed that, again given d , the factorization of x^3+d can be simplified by a precomputation in $\mathbb{Z}[\sqrt[3]{d}]$, though the complexity of the precomputation depends unpredictably on d via the arithmetic of the number field $\mathbb{Q}(\sqrt[3]{d})$; this approach was implemented in [HBLR]. Note that in effect these methods find rational points near the Fermat cubic that are close to the tangents to the curve at its inflection points — the same tangents that demand special care in our algorithm. A further variation which we suggested in 1996 is to use the factorization

$$z^3 - d = x^3 + y^3 = (x+y)(x^2 + xy + y^2) \quad (21)$$

as follows: fix $x+y$, solve for $z \bmod x+y$, and try each of the resulting values of z . Here we only find solutions with z , not x , bounded by N , but the advantage is that factoring costs are greatly diminished. To find all cube roots of $d \bmod x+y$ requires factoring $x+y$, a number of size N rather than N^3 ; and with enough space to set up a sieve the factorization can be avoided entirely. In 1999, Eric Pine, Kim Yarbrough, Wayne Tarrant and Michael Beck, all graduate students at the University of Georgia, took up this suggestion, choosing $d = 30$, and found the first solution:

$$30 = 2220422932^3 - 283059965^3 - 2218888517^3 \quad (22)$$

We announced our new algorithm in the same 1996 posting to the **NMBRTHRY** mailing list, together with results of a search for solutions with $z < 10^7$ and $|d| < 10^3$. We did our search in **gp**, making our computation easy to program (since **gp** already provides multiprecision arithmetic and lattice reduction) but far from optimally efficient. In 1999, unaware of the work of the Georgia group, we asked Dan J. Bernstein for an efficient implementation. He soon wrote a C program that found all solutions with $z < 3 \cdot 10^9$ and $|d| < 10^4$, including (22) and many others. Several values of d had not been previously represented as the sum of three cubes. Detailed results and analysis will appear elsewhere. As usual, since we are interested in small d , not all $d \ll N$, the improvement by a factor $N^{1/13}$ should apply here as well to find all cases of $|z^3 - y^3 - x^3| \ll z^{3/13}$ with $z < N$, but we have not attempted to implement such a computation.

3.3 Miscellaneous Examples

Trinomial Units. One sometimes sees in Olympiad-style mathematics contests the question “Is $z^{1/3}$ greater or smaller than $x^{1/3} + y^{1/3}$?” for some specific positive integers x, y, z . Of course this is a challenge only when the sign of the difference $u_3 := z^{1/3} - (x^{1/3} + y^{1/3})$ cannot be determined by inspection. In some cases the question be settled by applying classical inequalities; for instance if $a > b > 0$ then $(a+b)^{1/3} + (a-b)^{1/3} < 2a^{1/3}$ by convexity of the cube root.

The general solution is to compute the norm of $u_3/z^{1/3}$, an algebraic number of degree 9 none of whose other conjugates is real unless $x = y$. We find that u_3 has the same sign as

$$\mathbf{N}(x, y, z) := (z - y - x)^3 - 27xyz. \quad (23)$$

Moreover, given the size of x, y, z , the smaller $\mathbf{N}(x, y, z)$ is, the nearer u_3 will be to 0. In particular, we would like to have $\mathbf{N}(x, y, z) = \pm 1$, which would make the algebraic integer u_3 a unit. Thus again we seek rational points close to a plane cubic curve, here $\mathbf{N}(x, y, z) = 0$. This time the curve is rational: by construction, it is parametrized by $(x : y : z) = (t^3 : (1-t)^3 : 1)$. It is thus not smooth, but its only singularity is the isolated point $(x : y : z) = (1 : -1 : -1)$ (geometrically a node with complex conjugate tangents), which does not affect our algorithm. The three rational points of inflection at $xyz = z - y - x = 0$ do affect our algorithm, but fortunately we are not interested in the points on their tangent lines, since those are the points with $xyz = 0$. We thus restrict our attention to the portion of the curve with $x/z, y/z > 1/N$, i.e. with $t \gg N^{-1/3}$ and $1-t \gg N^{-1/3}$ in the rational parametrization. This takes us far enough from the inflection points that they cause us no difficulty.

The situation is now much the same as for $z^3 - y^3 - x^3 = d$. We expect the number of solutions of $\mathbf{N}(x, y, z) = d$ of height up to N to be proportional to $\log N$ times a product of local factors $g_p(d)$. The only local factor that can vanish is $g_3(d)$, which is nonzero if and only if $9|d$ or $d \equiv \pm 1 \pmod{9}$. We henceforth assume that d is in one of these congruence classes. We can then check whether $\prod_p g_p$ converges by comparing it with the L -series of the projective cubic surface $\mathbf{N}(x, y, z) = dt^3$. This in turn depends on the Galois structure of the Néron-Severi group of the surface, which can be determined from the action of Galois on the lines on that cubic surface, as explained in [W1]. We must be careful here because, unlike $x^3 + y^3 + z^3 = dt^3$, the surfaces $\mathbf{N}(x, y, z) = dt^3$ are not smooth: each has an A_2 singularity at $(x : y : z : t) = (1 : -1 : -1 : 0)$. Thus each has, not 27 lines as usual, but 15, of which 6 go through the singularity; see [BW]. Explicitly, these are the preimages under the projection to $(x : y : z)$ of the three coordinate axes and the two tangents to the curve at $(1 : -1 : -1)$. We conclude that, as with (19), $\prod_p g_p(d)$ converges unless d is a cube. So we expect the number of unparametrized solutions of height $\leq N$ to grow as $\log N$, except when d is a cube, when it should grow faster, albeit still as a power of $\log N$ — perhaps $\log^3 N$, by analogy with Manin's conjecture for cubic surfaces.

Unlike the case of (19), we know of no solutions of $\mathbf{N}(x, y, z) = \pm 1$ in nonconstant polynomials $x, y, z \in \mathbb{Z}[t]$, other than the trivial ones with $xyz = 0$. Nevertheless we can find infinitely many nontrivial integer solutions parametrized by Fermat-Pell equations, and thus show that the number of solutions of height $\leq N$ is $\gg \log N$. There are several ways to do this. In 1982 we found a somewhat complicated route to such a parametrization, obtaining a family of solutions starting with $\mathbf{N}(16948, 31226, 186919) = -1$. The details may be found in the pages of [CM]. Many years later, we observed that a simpler approach is to factor

$$\mathbf{N}(x, y, z) = \pm 1 \text{ as}$$

$$27xyz = (z - y - x)^3 \mp 1 = (z - y - x \mp 1)[(z - y - x)^2 \pm (z - y - x) + 1]. \quad (24)$$

For each $r \in \mathbb{Q}^*$, we obtain a conic curve C_r by setting $(z - y - x \mp 1) = rx$ in (24). This can be viewed geometrically as follows: the affine surface $\mathbf{N}(x, y, z) = \pm 1$ contains the line $x = (z - y - x \mp 1) = 0$; thus the intersection of the surface with any plane $(z - y - x \mp 1) = rx$ containing that line is the union of the line and some residual conic, which is our C_r . Likewise we could start from the line $z = (z - y - x \mp 1) = 0$ and intersect it with a variable plane $(z - y - x \mp 1) = rz$. For many choices of r , one of these conics is a hyperbola with infinitely many integral points parametrized by a Fermat-Pell equation.

In retrospect this approach to $\mathbf{N}(x, y, z) = \pm 1$, in which we fiber an affine surface by conics that may be regarded as principal homogeneous spaces for Fermat-Pell equations, seems a remarkable premonition of our later analysis [E1] of the projective quartic surface $A^4 + B^4 + C^4 = D^4$ via a fibration by genus-1 curves (principal homogeneous spaces for elliptic curves). In both cases the approach finds infinitely many solutions but does not readily lend itself to efficiently finding all solutions of height $\leq N$. Again a later computation found that the solution that was discovered first, because it lies on the first fiber that could contain a solution, is not the one of smallest height. We used our algorithm to find all small values of $\mathbf{N}(x, y, z)$ with $0 < x, y, z \leq 10^6$. We found that the smallest solution of $\mathbf{N}(x, y, z) = \pm 1$ is $(14, 84, 313)$ of norm $+1$, followed by $(6818, 4996, 46879)$, $(20388, 4881, 86830)$, and $(2742, 32540, 96843)$ each of norm -1 , the known $(16948, 31226, 186919)$, and $(3408, 182899, 370338)$ of norm $+1$, with no further solutions up to 10^6 . We also found several primitive solutions of $\mathbf{N}(x, y, z) = \pm 8$ and a few sporadic examples with d small but not a cube, which could not have been obtained at all using the factorization trick; the smallest of these are

$$\mathbf{N}(204, 115327, 162434) = 17, \quad \mathbf{N}(650, 1425, 7899) = 26. \quad (25)$$

The $\mathbf{N}(x, y, z) = 17$ solution yields a disappointingly large value of u_3 because the conjugates $z^{1/3} - y^{1/3} - e^{\pm 2\pi i/3} x^{1/3}$ are smaller than usual. An unexpected result — since the identity (10) cannot be used with exponents < 1 — was a polynomial solution of $\mathbf{N}(x, y, z) = 108$, namely $(4, y(t), -y(-1-t))$ where $y(t) = 4t^3 - 6t + 3$. We can write this symmetrically as $(8, g(t), -g(-t))$ where $g(t) = 8t^3 - 12t - 6t + 11$, a cubic polynomial determined up to scaling by the condition that the Laurent expansion at infinity of $(g(t))^{1/3}$ have vanishing t^{-2} and t^{-4} terms. In this form, $\mathbf{N}(x, y, z)$ is the larger constant $864 = 2^3 108$, but with the bonus that x is a cube so u_3 involves one fewer surd; for instance, taking $t = 7$ we find that $\sqrt[3]{3279}$ is smaller than $2 + 5\sqrt[3]{17}$ by less than $3.75 \cdot 10^{-7}$. In this family, as with the first example in (25), u_3 is of order z^{-2} , not $z^{-8/3}$, because two of the conjugates of u_3 are $O(1)$.

A similar investigation of $u_4 := z^{1/4} - (x^{1/4} + y^{1/4})$ was not as productive, perhaps not surprisingly since there are no arithmetic reasons to expect many nonzero small examples. For the record, the smallest $z^{11/4}|u_4|$ value found for

$z < 10^6$ was $0.365+$ for $(x, y, z) = (241, 691, 6759)$, while the smallest $|u_4|$ in that range was $(3.23-) \cdot 10^{-16}$ for $(37792, 36109, 591093)$.

The π -th Fermat Curve. To illustrate our algorithm also for non-algebraic curves, we chose to apply it to the Fermat curve of exponent π . Since π exceeds 3, but only slightly, we expected that $|z^\pi - y^\pi - x^\pi|$ achieves a global minimum over all x, y, z with $0 < x \leq y < z$ but that the minimum might involve numbers of several digits. We were rewarded with the example

$$2063^\pi + 8093^\pi - 8128^\pi = 0.019369- = 8128^{\pi-3}/(184.75+), \quad (26)$$

which seems likely to be the minimum of $|z^\pi - y^\pi - x^\pi|$ over all positive integers x, y, z . At any rate, according to our computations it is the smallest with $z \leq 10^6$. The ratio $184.75+$ is also the largest in that range, though there is also

$$1198^\pi + 4628^\pi - 4649^\pi = -(0.04949+) = -4649^{\pi-3}/(66.794+). \quad (27)$$

It will probably be a long time before the question of the minimality of (26) is settled; a weaker but still intractable conjecture is that there are only finitely many integer solutions of $|z^\pi - y^\pi - x^\pi| < 1$.

The Klein Quartic. All our examples so far were Fermat curves, even though some had unusual exponents $1/3, 1/4, \pi$. Probably the best-known projective plane curve that is not a Fermat curve is the Klein quartic $K(X, Y, Z) = 0$, where

$$K(X, Y, Z) := X^3Y + Y^3Z + Z^3X. \quad (28)$$

We used our algorithm to search for small values of $K(x, y, z)$. By symmetry we may assume $\max(x, y, z) = z$. We are then seeking rational points near a segment of a plane curve with a single inflection point, at $x = y = 0$. The tangent $x = 0$ at this point accounts for the obvious family $(0, 1, z)$ with $K(x, y, z) = z$. Our computation up to height 10^6 quickly revealed a less obvious family, $K(1, -t^2, t^3) = -t^2$, with $K(x, y, z)$ growing even more slowly than the height. As usual we also found sporadic examples, though here (as with several other cases we have already seen such as the Fermat quintic) the best ones are small enough that our algorithm was not needed to locate them:

$$\begin{aligned} K(1421, -1057, 1501) &= -49, \\ K(7211, -8381, 11010) &= -121, \\ K(-1550, 11817, 32615) &= 245, \end{aligned} \quad (29)$$

with $z/|K(x, y, z)| = 30.6, 91.0, 133.1$ respectively. The largest $z/|K(x, y, z)|$ found with $z \in [10^5, 10^6]$ off the singular cubic $y^3 + x^2z = 0$ was 6.756+, from $K(-7871, 175577, 829244) = 122741$.

4 Hall's Conjecture

4.1 Review of Hall's Conjecture

By *Hall's conjecture* we mean the following assertion: if x, y are positive integers such that

$$k := x^3 - y^2 \tag{30}$$

is nonzero (equivalently, such that $(x, y) \neq (t^2, t^3)$), then

$$|k| \gg_{\epsilon} x^{1/2-\epsilon}. \tag{31}$$

(While this accords with current usage, it is not exactly what Hall originally wrote: as F. Beukers points out, Hall [H] conjectured $|k| \gg x^{1/2}$, a stronger statement which is probably false — the usual heuristic suggests that there are at least $(\delta + o(1)) \log X$ cases of $0 < |k| < \delta \sqrt{x}$ with $x < X$ — but unlikely to be soon disproved. See also [BCHS] for the early history of this conjecture.) Among several equivalent forms of (31) we note the conjecture that the discriminant of an elliptic curve over \mathbb{Q} in its standard minimal form has absolute value $\gg_{\epsilon} |a_4|^{1/2-\epsilon}$. Known lower bounds on $|k|$ are much weaker than (31). By Siegel's theorem on the finiteness of integer points on elliptic curves, each nonzero $k \in \mathbb{Z}$ occurs finitely many times as $x^3 - y^2$, so $|k| \rightarrow \infty$ as $x \rightarrow \infty$. Siegel's proof is ineffective and thus says nothing about how fast $|k|$ must grow with x . Starting with Baker's method, effective bounds have become available, but they are still very weak. For instance, it is not yet possible to prove for any $\theta > 0$ that $|k| \gg x^{\theta}$.

Hall's conjecture is now recognized as an important special case of the Masser-Oesterlé ABC conjecture [O] (see also [L]). Thus its analogue over function fields is known to be true by Mason's theorem [M3]. In the special case of Hall's conjecture for polynomials $x(t), y(t)$, the fact that $x^3 - y^2$ is either zero or has degree $> \frac{1}{2} \deg(x)$ was proved some twenty years earlier by Davenport [D2] in response to a question raised in [BCHS]. As in [E2] it follows that the conjecture cannot be disproved by a polynomial parametrization, and indeed in any polynomial family $(x(t), y(t)|t \in \mathbb{Z})$ we must have $k \gg x^{\theta}$ with $\theta > 1/2$. One does better with solutions parametrized by Fermat-Pell equations, i.e. $x, y \in \mathbb{Z}[t, \sqrt{at^2 + bt + c}]$ for some $a, b, c \in \mathbb{Z}$ such that $u^2 = at^2 + bt + c$ has infinitely many solutions. The function field $\mathbb{Q}(t, \sqrt{at^2 + bt + c})$ is then still rational, so the Davenport-Mason inequality again holds, but since now there are two places at infinity one can have $x^3 - y^2$ of degree exactly $\frac{1}{2} \deg(x)$, and thus attain $\theta = 1/2$. The existence of a single such family (exhibited below) shows that the exponent in (31) cannot be raised above 1/2. The fact that one cannot reduce θ below 1/2 in this way was again observed in [E2] in the more general context of the ABC conjecture. This fact lends some credence to that conjecture, and thus to its special case (31); this contrasts with the situation for $|z^n - y^n - x^n|$, where there is no reason why some polynomial or Pell family might not do better than the $z^{n-3-\epsilon}$ expected by probabilistic heuristics, and indeed we found such families for some choices of n .

We next digress to say some more on polynomial and Fermat-Pell families that attain the Davenport-Mason bound, both because they are of independent interest and because families of both kinds appear in our numerical results. In either case $x^3 - y^2 = k$ is an identity in a genus-zero function field, namely $\mathbb{Q}(t)$ in the polynomial case and $\mathbb{Q}(t, \sqrt{at^2 + bt + c})$ in the Fermat-Pell case. Let x, y have degrees $2m, 3m$ respectively, and suppose k has the smallest degree possible, i.e. $m+1$ in the polynomial case and m for Fermat-Pell. Then $f := x^3/y^2$ is a rational function of degree $6m$ or $12m$ on \mathbb{P}^1 ramified only above $0, 1, \infty$. The Riemann existence theorem provides infinitely many such functions $f = x^3/y^2$ in $\mathbb{C}(t)$; this answers the first part of the question raised in [BCHS, p.68]. The second part concerns solutions over \mathbb{R} , and can probably be settled by adding data on complex conjugation to the branched covering. But we are most interested in the third part of the question, in which f must have rational coefficients. Given any one $(x(t), y(t))$, we may trivially obtain others of the form $(x', y') = (\lambda^2 x(t'), \lambda^3 y(t'))$ where $t' = at + b$ in the polynomial case, and $t' \in \mathbb{Q}[t]$ with $\sqrt{at'^2 + bt' + c}/\sqrt{at^2 + bt + c} \in \mathbb{Q}[t]$ in the Fermat-Pell case. If we regard such (x', y') and (x, y) as equivalent, only a handful of examples over \mathbb{Q} are known, and there may well be no others. We next list representatives of the known examples.

In the polynomial case, all known examples have $m \leq 5$. For $m=1$, translation and scaling brings any quadratic $x(t)$ to the form $t^2 + 2a$, and then $y = t^3 + 3at$ and $k = 3a^2t^2 + 8a^3$. Necessarily $a \neq 0$, and all such examples are “twists” of each other, becoming isomorphic over $\bar{\mathbb{Q}}$ if not over \mathbb{Q} . Note that x^3/y^2 is a degree-3 function of t^2 with a triple zero. This function occurs for instance as the cover of the modular curve $X(1)$ by $X_0(2)$. For $m=2$ we again find that the solution is unique up to twist: $x = t^4 + 4at$, $y = t^6 + 6at^3 + 6a^2$, and $k = -8a^3t^3 - 36a^4$. This time x^3/y^2 is a degree-4 function of t^3 , whose ramification identifies it with the modular cover $X_0(3) \rightarrow X(1)$. Birch found examples of (x, y, k) with $m=3, 5$ and included them in a 29.ix.1961 letter to Chowla; they are reported in [BCHS]:

$$\left(36t^6 + 24t^4 + 10t^2 + 1, 216t^9 + 216t^7 + 126t^5 + 35t^3 + \frac{21}{4}t, \frac{9}{2}t^4 + \frac{39}{16}t^2 + 1 \right), \quad (32)$$

and

$$\left(\frac{t}{9}(t^9 + 6t^6 + 15t^3 + 12), \frac{t^{15}}{27} + \frac{t^{12} + 4t^9 + 8t^6}{3} + \frac{5t^3 + 1}{2}, -\frac{3t^6 + 14t^3 + 27}{108} \right). \quad (33)$$

These yield integer solutions if t is a multiple of 4 in (32) or congruent to 3 mod 6 in (33). As noted in [BCHS], the second example provides infinitely many integer solutions of $|x^3 - y^2| \ll x^{3/5}$; moreover, for this choice of twist, the leading coefficient of $k(t)$ is small enough that $|x^3 - y^2|$ is even a respectably small multiple of $x^{1/2}$ for the first few specializations of t . The maps $f = x^3/y^2$ associated with Birch's polynomials both have interesting Galois groups. For (32), f is a degree-9 function of t^2 whose Galois group is $\mathrm{PSL}_2(\mathbb{F}_8)$ over $\mathbb{C}(t^2)$ and $\mathrm{Aut}(\mathrm{PSL}_2(\mathbb{F}_8))$ over $\mathbb{Q}(t^2)$; the Galois closure is the Fricke-Macbeath curve [F,M1]. For (33), f is a degree-10 function of t^3 whose Galois group is $\mathrm{PSL}_2(\mathbb{F}_9)$. These groups and curves do not arise in connection with classical modular curves, but they

can be identified with certain Shimura modular curves, most naturally those associated with the $(2, 3, 7)$ and $(2, 3, 8)$ arithmetic triangle groups (see for instance [T,E5]). Hall [H, p.185] gives an example with $m = 4$:

$$x = 4(t^8 + 6t^7 + 21t^6 + 50t^5 + 86t^4 + 114t^3 + 109t^2 + 74t + 28); \quad (34)$$

In August 1998 I announced a new example with $m = 5$ (its computation will be explained elsewhere):

$$x = t^{10} + 2t^9 + 33t^8 + 12t^7 + 378t^6 - 336t^5 + 2862t^4 - 2652t^3 + 14397t^2 - 9922t + 18553. \quad (35)$$

In both cases (as with all the other (x, y, k) examples), y is obtained by truncating the Laurent expansion at infinity of $x^{3/2}$ after the constant term. Neither (34) nor (35) yields an interesting Galois group: the Galois groups of x^3/y^2 are Alt_{24} and Sym_{30} respectively. While (35), like (33), must yield infinitely many integer solutions of $|x^3 - y^2| \ll x^{3/5}$, the leading coefficient of k in (35) makes the implied constant much larger, and none of these solutions will appear in our list of small values of $|x^3 - y^2|$. The question, raised in [BCHS], whether there are any $x, y, k \in \mathbb{Q}[t]$ of degrees $2m, 3m, m+1$ with $m > 5$, remains unsolved.

For Fermat-Pell families, the list is even shorter: all known examples are equivalent, and come from the identity

$$(t^2 + 10t + 5)^3 - (t^2 + 22t + 125)(t^2 + 4t - 1)^2 = 1728t. \quad (36)$$

Here y is a multiple of $\sqrt{at^2 + bt + c}$, so f factors as a map of degree 6 composed with the double cover of $\mathbb{Q}(t)$ by $\mathbb{Q}(t, \sqrt{at^2 + bt + c})$. We noted in [E4, p.49] that the resulting degree-6 map $f = x^3/y^2 : \mathbb{P}^1 \rightarrow \mathbb{P}^1$ is the cover $X_0(5) \rightarrow X(1)$ of classical modular curves. Thus the elliptic curves of low discriminant coming from the identity (36) all admit a rational 5-isogeny. Each Fermat-Pell family obtained from (36) by specifying the class of $t^2 + 22t + 125 \bmod \mathbb{Q}^{*2}$ yields $k \sim Cx^{1/2}$ for some nonzero C . The smallest such C is $5^{-5/2}54 = .96598\dots$, obtained by Danilov [D1] by substituting $125(2t - 1)$ for t in (36) and dividing by 20^3 :

$$(5^5t^2 - 3000t + 719)^3 - (5^3t^2 - 114t + 26)(5^6t^2 - 5^3 123t + 3781)^2 = 27(2t - 1). \quad (37)$$

The factor $5^3t^2 - 114t + 26$ is a square for $t = -5$, and thus for infinitely many t . The first case $t = -5$ of this yields the elliptic curve of discriminant -11 labeled 11-A2(C) in Cremona's table [C2]; it is known that the isogeny class of this curve provides the examples with minimal conductor of a rational 5-isogeny, and indeed of an elliptic curve over \mathbb{Q} .

4.2 The New Algorithm

To obtain numerical data with which to compare Hall's conjecture, we want to find all small nonzero values of $|x^3 - y^2|$ with $x, y \in \mathbb{Z}$ and $x \leq X$. So that we can compare our algorithm with other approaches we briefly review previous work on this problem.

The most direct approach is to simply compute for each $x \leq X$ the integer y closest to $x^{3/2}$. Since $x^{3/2}$ varies smoothly with x , this can be done quite efficiently, but clearly must take at least time proportional to X . This is essentially what Hall did in [H], with $X = 7 \cdot 10^8$; some three decades later, faster computers make larger X feasible, and indeed Frits Beukers reports in an Aug. 1998 e-mail that he performed such a computation for $X = 10^{12}$. But this is probably close to the practical limit with today's technology, and at any rate this direct approach is superseded by the $X^{1/2} \log^{O(1)} X$ algorithm described below.

A fundamentally different approach is taken in [GPZ]: for each nonzero $k \in [-K, K]$, investigate the arithmetic of the elliptic curve $E_k : y^2 = x^3 - k$, and use effective bounds on integral points to find all integer solutions of $x^3 - y^2 = k$. In [GPZ], Gebel, Pethö, and Zimmer did most of this work for $K = 10^5$, except for a few values of k , for which they could not find a generator for $E_k(\mathbb{Q})$; Wildanger later showed in his doctoral thesis [W2] that none of these E_k has an integral point, thus completing the computation of integer solutions of $0 < |x^3 - y^2| < 10^5$. It is not clear even heuristically how this method compares with other approaches. It is the only approach used thus far that will provably find all solutions with $|k| \leq K$. (The recent proof of the modularity conjecture means that Cremona's algorithms [C2] yield another such approach, but to my knowledge it has not been used to solve $x^3 - y^2 = k$.) Assuming Hall's conjecture, $|k| \leq K$ is equivalent to $x \ll_{\epsilon} K^{2+\epsilon}$, but this begs the question of the constant implied in “ \ll ”. Neither do we know how to estimate the average work required to find all integer points on a curve E_k . It may be reasonable to guess that this average work is proportional to $K^{c+o(1)}$ for some $c > 0$. (This estimate certainly holds for Cremona's algorithms.) The total work would then be $K^{1+c+o(1)}$. Under Hall's conjecture, this is equivalent to $X^{(1+c)/2+o(1)}$, so strictly worse (modulo an unknown implied constant) than our $X^{1/2} \log^{O(1)} X$ algorithm, though perhaps better than a direct search, depending on whether $c < 1$.

We noted already that the direct search can exploit the smoothness of the function $x^{3/2}$. We can try to take further advantage of this by mimicking our approach to rational approximation of curves: surround the segment $x < X$ of the semicubical parabola $y = x^{3/2}$ by a union of parallelograms each of area $O(1)$, and use lattice reduction to quickly find all integer points in each parallelogram. This does give an asymptotic improvement, though a small one: the parallelogram containing a point $(x, x^{3/2})$ has length $\gg x^{1/6}$, so the computational cost is reduced by at most $X^{1/6}$, to $X^{5/6} \log^{O(1)} X$.

We reduce the exponent of X from 1 or $5/6$ to $1/2$ by a more radical reorganization of the computation that lets us apply lattice reduction more efficiently. More generally, for each positive $c \in \mathbb{Q}$ we can find all cases of $0 < |cx^3 - y^2| \ll x$ in time $O_c(X^{1/2} \log^{O(1)} X)$. All choices of c are essentially equivalent: we get from one to the other by scaling x, y and imposing congruence conditions on them. The most convenient choice of c turns out to be $4/3$. We thus show how to solve $0 < |4x^3 - 3y^2| \ll x$; the cases relevant to Hall's conjecture are those with $3|x$ and $6|y$, when $(4x^3 - 3y^2)/108 = (x/3)^3 - (y/6)^2$.

We begin as in [H] by approximating x, y by (multiples of) a square and a cube. Any positive integer x may be written uniquely as

$$x = 3\zeta^2 + \eta \quad \text{with} \quad \eta, \zeta \in \mathbb{Z}, \zeta > 0, \eta \in (-3\zeta, 3\zeta]. \quad (38)$$

Then

$$(4x^3/3)^{1/2} = 6\zeta^3 + 3\eta\zeta + \frac{1}{4} \frac{\eta^2}{\zeta} - \frac{1}{72} \left(\frac{\eta}{\zeta} \right)^3 + O(1/\zeta). \quad (39)$$

We thus write

$$y = 6\zeta^3 + 3\eta\zeta + \xi \quad (40)$$

with $\xi \ll \zeta$. More precisely, if

$$\eta = \beta\zeta \quad (41)$$

Then $\beta \in (-3, 3]$, and $|4x^3 - 3y^2| \ll x$ if and only if

$$\xi = \frac{\eta^2}{4\zeta} - \frac{1}{72}\beta^3 + O(1/\zeta). \quad (42)$$

At this point, Hall [H] imposes the assumption $\beta \ll \xi^{-1/5}$. We allow an arbitrary $\beta \in (-3, 3]$ and approximate it within $O(X^{-1/2})$ by one of $O(X^{1/2})$ evenly spaced points in that interval. Suppose, then, that b is one of those points. We approximate (42) by a linear combination of $\zeta, \eta - b\zeta$, and 1:

$$\xi = \frac{b^2}{4}\zeta + \frac{b}{2}(\eta - b\zeta) - \frac{b^3}{72} + O(1/\zeta) = -\frac{b^2}{4}\zeta + \frac{b}{2}\eta - \frac{b^3}{72} + O(1/\zeta). \quad (43)$$

We now assume that $\zeta \gg X^{1/2}$, for instance by requiring that $x > X/4$; repeating the computation with X replaced by $X/4, X/16, X/64, \dots$ will then cover the entire range $x \leq X$, and if we can cover $(X/4, X]$ in time $O(x^{1/2} \log^{O(1)} X)$ then the same is true of $[1, X]$. Under the assumption $x \in (X/4, X]$, we have the following constraints on ξ, η, ζ :

$$\zeta \ll X^{1/2}, \quad \eta - b\zeta \ll 1, \quad (44)$$

and

$$\xi + \frac{b^2}{4}\zeta - \frac{b}{2}\eta + \frac{b^3}{72} \ll X^{-1/2}. \quad (45)$$

We are thus in a familiar situation: we seek all the integral points in $O(X^{1/2})$ parallelepipeds, each of volume $O(1)$. The term $b^3/72$ in (45) means that the parallelepipeds are no longer centered at the origin, but this causes no difficulty — indeed we already dealt with off-center parallelepipeds in the practical implementation of our algorithm for finding rational points near curves. So again we linearly transform each parallelepiped to a cube and obtain a lattice reduction problem; if these lattices were randomly distributed among three-dimensional lattices, we would almost certainly have only $O(X^{1/2})$ points to try, and would thus find all solutions of $0 < |4x^3 - 3y^2| \ll x$ with $x \leq X$ in time $O(X^{1/2} \log^{O(1)} X)$.

In fact it turns out that in this case our lattices are *not* equidistributed: they all lie in a 2-dimensional subspace of the 5-dimensional moduli space of lattices in \mathbb{R}^3 . This gives rise to both a minor annoyance and a major advantage. The bad news is that we cannot expect our lattices to have on average $O(1)$ vectors of norm $\ll 1$; but this annoyance is minor because the actual average is proportional to $\log X$ and thus can be absorbed into the $\log^{O(1)} X$ factor. The good news is that we understand our special lattices well enough to actually prove results that are only heuristic for rational points near curves.

The key is that in each case our lattice is a *symmetric square* of a lattice in \mathbb{R}^2 . By this we mean the following. Recall that the symmetric square of a 2-dimensional vector space V is the 3-dimensional vector space $\text{Sym}^2 V$ consisting of symmetric tensors in $V \otimes V$. Since $\text{Sym}^2 V$ is defined naturally in terms of V , any linear transformation of V yields a linear transformation of $\text{Sym}^2 V$. We thus have a homomorphism $\text{Sym}^2 : \text{GL}_2 \rightarrow \text{GL}_3$. To give this map explicitly we choose a basis (e_1, e_2) for V , and use the basis $(e_1 \otimes e_1, (e_1 \otimes e_2 + e_2 \otimes e_1)/2, e_2 \otimes e_2)$ for $\text{Sym}^2 V$. We then calculate that

$$\text{Sym}^2 \begin{pmatrix} p & q \\ r & s \end{pmatrix} = \begin{pmatrix} p^2 & pq & q^2 \\ 2pr & ps + qr & 2qs \\ r^2 & rs & s^2 \end{pmatrix}. \quad (46)$$

Over any field, $\text{Sym}^2(\text{SL}_2)$ is contained in the subgroup of SL_3 preserving the discriminant form $4a_1a_3 - a_2^2$ on $\text{Sym}^2(V)$; if we worked over an algebraically closed field, that subgroup would coincide with $\text{Sym}^2(\text{SL}_2)$. Now (44,45) mean that the column vector $v = (\xi, \eta, \zeta) \in \mathbb{Z}^3$ satisfies $\|M_b v - u_b\| \ll 1$ where $u_b := (0, 0, -b^3/72)$ and

$$M_b := \begin{pmatrix} 0 & 0 & X^{-1/2} \\ 0 & 1 & -b \\ X^{1/2} & -X^{1/2}b/2 & X^{1/2}b^2/4 \end{pmatrix} = \text{Sym}^2 \begin{pmatrix} 0 & X^{-1/4} \\ X^{1/4} & X^{1/4}b/2 \end{pmatrix}. \quad (47)$$

This is why we went after $4x^3 - 3y^2$ rather than pursuing $x^3 - y^2$ directly: an analogous approach to $x^3 - y^2$ would yield a matrix that is still a symmetric square but with respect to a different basis, requiring a definition of Sym^2 with fractional coefficients and complicating the lattice reduction. Note that the quadratic form $4\xi\zeta - \eta^2$ preserved by $M_b \in \text{Sym}^2(\text{SL}_2)$ is already visible in (42).

Our algorithm, then, is as follows. For each of our $O(X^{1/2})$ choices of b , calculate the matrix

$$N_b := \begin{pmatrix} 0 & X^{-1/4} \\ X^{1/4} & X^{1/4}b/2 \end{pmatrix} \quad (48)$$

with $M_b = \text{Sym}^2 = N_b$. Use lattice reduction to find a matrix $K_b \in \text{GL}_2(\mathbb{Z})$ such that $N_b K_b$ is as small as possible. Then

$$M'_b := \text{Sym}^2(N_b K_b) = \text{Sym}^2 N_b \text{Sym}^2 K_b = M_b \text{Sym}^2 K_b. \quad (49)$$

is small too. Let $L_b = \text{Sym}^2 K_b \in \text{GL}_3(\mathbb{Z})$. Then $M_b v = M'_b L_b^{-1} v$. Find a box containing all $w \in \mathbb{Z}^3$ such that $\|M'_b w - u_b\| \ll 1$. For each w in the box,

compute $v = L_b w$ and check whether the resulting x, y satisfy $x \in (X/4, X]$ and $0 < |4x^3 - 3y^2| \ll x$; if they do, output x (and check whether $3|x$ and $6|y$ to determine whether this solution also yields a small value of $x^3 - y^2$). This is easier than our usual algorithm because we are reducing a lattice in \mathbb{R}^2 rather than \mathbb{R}^3 , which in our case amounts to calculating the continued fraction of $b/X^{1/2}$. Moreover, the computational cost of the algorithm can be bounded rigorously: M'_b will only be large if $b/X^{1/2}$ is close to a rational number with numerator and denominator $\ll X^{1/4}$, and the effect of such a close rational approximation is easy to determine. Summing over all rationals of height $\ll X^{1/4}$ we find that the total number of candidate vectors v is $\ll X^{1/2} \log X$, and thus that the computation takes time $O(X^{1/2} \log^{O(1)} X)$ as claimed.

Note that the $X^{1/2} \log X$ bound also has the following consequence: there are $\ll X^{1/2} \log X$ solutions of $|x^3 - y^2| \ll \sqrt{x}$ with $x \leq X$. Moreover, if C is large enough, we can deduce from this analysis that there are $\gg X^{1/2}$ solutions of $0 < |x^3 - y^2| < Cx$ with $x \in [X/2, X]$. More generally, we show that for each positive $c \in \mathbb{Q}$ there exists C such that for each $r \in \mathbb{R}/\mathbb{Z}$ and $d > 1$ there are at most $CdX^{1/2} \log X$ solutions of $|(cx^3)^{1/2} - (y + r)| < dX^{-1/2}$ with $x, y \in \mathbb{Z}$ and $x < X$; and, given c as above and any $\theta \in [0, 1)$, there exists C_0 such that for any $r \in \mathbb{R}/\mathbb{Z}$ there are $\gg X^{1/2}$ solutions of $|(cx^3)^{1/2} - (y + r)| < C_0 X^{-1/2}$ with $x, y \in \mathbb{Z}$ and $x \in [\theta X, X]$. The constants C, C_0 depend effectively on c, θ . These results improve considerably on results in this direction available from general exponential-sum techniques for proving uniform distribution mod 1. The detailed proofs of our claims in this paragraph will appear elsewhere.

4.3 Numerical Results

We have implemented our algorithm in a C program using 64-bit integer arithmetic, again replacing each $O(\dots)$ and \ll by explicit bounds, and searched for all solutions of $0 < |4x^3 - 3y^2| < 200x^{1/2}$ with $4 \cdot 10^6 < x < 3 \cdot 10^{18}$. The range $x < 10^{10}$ was covered by a direct search, the overlap $[4 \cdot 10^6, 10^{10}]$ being used as a check on the computation. The code was processed with an optimizing compiler and ran for three weeks during the summer of 1998 on a Sun Sparcstation Ultra 1. As a corollary we obtained all cases of $0 < |x^3 - y^2| < \frac{1}{2}\sqrt{x}$ with $x < 10^{18}$. (With currently available hardware the same computation could easily finish in a few days; with parallelization it should be feasible to reach 10^{23} at least.) The next table lists, for each of the 25 solutions of $0 < |x^3 - y^2| < \frac{1}{2}\sqrt{x}$, the values of $k = x^3 - y^2$, x , and $r = x^{1/2}/|k|$. We need not list y , which is always the integer nearest to $x^{3/2}$. The explanation of the last two columns follows the table.

The “GPZ” column indicates whether the solution was among the 13 listed in [GPZ]. These are the solutions with $1 < |k| < 10^5$. Presumably the solution $2^3 - 3^2 = -1$ is not on that list because the elliptic curve $y^2 = x^3 + 1$ was already known to have rank 0 so Gebel, Pethö and Zimmer were not interested in it.

The #1 row is a new record, improving the previous record r by a factor of almost 10, whence the notation “!!”. Even row #2, marked “!”, has r larger than the old record which is row #3. Either of this suffices to refute Hall’s comment [H, p.175], repeated in [GPZ], that $r < 5$ seems to hold in all cases.

*: Obtained from row #1 by scaling (x, y, k) to $(2^2 x, 2^3 y, 2^6 k)$. This reduces r by a factor of 32, but $r = 46+$ in row #1 is large enough that even $r/32$ still exceeds the threshold of our table.

$P(t)$: Birch's polynomial family (33). This has $r = 12/t + O(t^{-4})$, so the only values of $t \equiv 3 \pmod{6}$ that appear on the $r > 1$ list are $t = \pm 3$ and ± 9 . Already in [BCHS, p.69] the specializations $t = \pm 3$ are noted as “striking special cases” of (33).

D: The first two cases of Danilov's family (37). The appearance of the larger of these was a welcome check on our computation.

Any threshold on r is of necessity arbitrary; the next solution has r just below our cutoff of 1: $(x, k, r) = (16544006443618, 4090263, 0.9944\dots)$.

#	k	x	r	GPZ?	Comments
1	1641843	5853886516781223	46.60		!!
2	30032270	38115991067861271	6.50		!
3	-1090	28187351	4.87	+	
4	-193234265	810574762403977064	4.66		
5	-17	5234	4.26	+	$P(-3)$
6	-225	20114	3.77	+	
7	-24	8158	3.76	+	$P(3)$
8	307	939787	3.16	+	
9	207	367806	2.93	+	
10	-28024	3790689201	2.20	+	
11	-117073	65589428378	2.19		
12	-4401169	53197086958290	1.66		
13	105077952	23415546067124892	1.46		*
14	-1	2	1.41		
15	-497218657	471477085999389882	1.38		
16	-14668	384242766	1.34	+	$P(-9)$
17	-14857	390620082	1.33	+	$P(9)$
18	-87002345	12813608766102806	1.30		
19	2767769	12438517260105	1.27		
20	-8569	110781386	1.23	+	
21	5190544	35495694227489	1.15		
22	-11492	154319269	1.08	+	
23	-618	421351	1.05	+	
24	548147655	322001299796379844	1.04		D
25	-297	93844	1.03	+	D

Acknowledgements

Richard K. Guy wrote the book [G] that first introduced me to many open problems in number theory including the Diophantine equations $x^3 + y^3 + z^3 = d$ [G, Prob. D5], and later brought me up to date on recent work on this problem. Dan J. Bernstein efficiently implemented my new algorithm for the problem. Alan Murray told me of the appearance of approximate integer solutions

of $x^{12} + y^{12} = z^{12}$ on *The Simpsons*. Frits Beukers and Franz Lemmermeyer filled gaps in my knowledge of earlier work concerning Hall's conjecture. Barry Mazur suggested that a method for locating points near a variety might also profitably be applied to finding points on the variety; this started me thinking in the direction that led to Theorems 2 through 4. Alf van der Poorten and Hugh Montgomery directed me to Bombieri and Pila's work [BP] concerning integral points on curves; Peter Sarnak put me in contact with Pila, who noted his more recent paper [P]; meanwhile Victor Miller alerted me to results announced by Roger Heath-Brown [HB2], who discussed his and Pila's work with me. Meanwhile, Michel Waldschmidt informed me of relevant results by Weierstrass and others collected in [M2, Chapter 3]. I thank them all for these contributions to the present paper.

Most of the numerical and symbolic computations reported here were carried out using the GP/PARI and MACSYMA packages.

This work was made possible in part by funding from the David and Lucile Packard Foundation.

References

- [B1] Bernstein, D.J.: Enumerating solutions to $p(a) + q(b) = r(c) + s(d)$. *Math. of Computation*, to appear.
- [B2] Bremner, A.: Sums of three cubes. Pages 87–91 in *Number Theory (Halifax, Nova Scotia, 1994)* (CMS Conf. Proc. 15), Providence: AMS, 1995.
- [BCHS] Birch, B.J., Chowla, S., Hall, M.Jr., Schinzel, A.: On the difference $x^3 - y^2$, *Norske Vid. Selsk. Forh.* **38** (1965), 65–69.
- [BP] Bombieri, E., Pila, J.: The number of integral points on arcs and ovals. *Duke Math. J.* **59** (1989) #2, 337–357.
- [BW] Bruce, J.W., Wall, C.T.C.: On the classification of cubic surfaces. *J. London Math. Soc.* (2) **19** (1979) #2, 245–256.
- [C1] Cohn, H.L.: *New Bounds on Sphere Packings*. Ph.D. thesis, Harvard 2000.
- [C2] Cremona, J.E.: *Algorithms for modular elliptic curves*. Cambridge University Press, 1992.
- [CM] *Crux Mathematicorum* **8** (1982).
- [CS] Conway, J.H., Sloane, N.J.A.: *Sphere Packings, Lattices and Groups*. New York: Springer 1993.
- [CV] Conn, W., Vaserstein, L.N.: On sums of integral cubes. Pages 285–294 in *The Rademacher legacy to mathematics* (University Park, 1992), *Contemp. Math.* **166**, AMS 1994.
- [D1] Danilov, L.V.: The Diophantine equation $x^3 - y^2 = k$ and Hall's conjecture, *Math. Notes Acad. Sci. USSR* **32** (1982), 617–618.
- [D2] Davenport, H.: On $f^3(t) - g^2(t)$, *Norske Vid. Selsk. Forh.* **38** (1965), 86–87.
- [E1] Elkies, N.D.: On $A^4 + B^4 + C^4 = D^4$, *Math. of Computation* **51** (Oct.88) #184, 825–835.
- [E2] Elkies, N.D.: ABC implies Mordell, *International Math. Research Notices* 1991 #7, 99–109.
- [E3] Elkies, N.D.: Heegner point computations. Pages 122–133 in *Algorithmic Number Theory* (Proceedings of ANTS-I; L.M. Adleman, M.-D. Huang, eds.; Berlin: Springer, 1994; Lecture Notes in Computer Science **877**).
- [E4] Elkies, N.D.: Elliptic and modular curves over finite fields and related computational issues. Pages 21–76 in *Computational Perspectives on Number Theory*:

- Proceedings of a Conference in Honor of A.O.L. Atkin* (D.A. Buell and J.T. Teitelbaum, eds.; AMS/International Press, 1998).
- [E5] Elkies, N.D.: Shimura curve computations. Pages 1–47 in *Algorithmic Number Theory* (Proceedings of ANTS-III; J. P. Buhler, ed.; Berlin: Springer, 1998; Lecture Notes in Computer Science **1423**).
- [F] Fricke, R.: Ueber eine einfache Gruppe von 504 Operationen, *Math. Ann.* **52** (1899), 321–339.
- [FH] Fulton, W., Harris, J.: *Representation Theory: A First Course*. New York: Springer, 1991 (GTM **129**).
- [GPZ] Gebel, J., Pethö, A., and Zimmer, H.G.: On Mordell's equation, *Compositio Math.* **110** (1998), 335–367.
- [G] Guy, R.K.: *Unsolved Problems in Number Theory*. New York: Springer, 1981.
- [H] Hall, M.: The Diophantine equation $x^3 - y^2 = k$. Pages 173–198 in *Computers in Number Theory* (A. Atkin, B. Birch, eds.; Academic Press, 1971).
- [HB1] Heath-Brown, D.R.: The density of zeros of forms for which weak approximation fails. *Math. of Computation* **59** (1992) #200, 613–623.
- [HB2] Heath-Brown, D.R.: The density of rational points on projective hypersurfaces. Preprint, 2000.
- [HBLR] Heath-Brown, D.R., Lioen, W.M., te Riele, H.J.J.: On solving the Diophantine equation $x^3 + y^3 + z^3 = k$ on a vector computer. *Math. of Computation* **61** (1993) #203, 235–244.
- [KK] Keller, W., Kulesz, L.: Courbes algébriques de genre 2 et 3 possédant de nombreux points rationnels. *C. R. Acad. Sci. Paris, Sér. I Math.* **321** (1995) #11, 1469–1472.
- [KTS] Koyama, K., Tsuruoka, U., Sekigawa, H.: On searching for solutions of the Diophantine equation $x^3 + y^3 + z^3 = n$. *Math. of Computation* **66** (1997) #218, 841–851.
- [L] Lang, S.: Old and new conjectured diophantine inequalities, *Bull. Amer. Math. Society* #23 (1990), 37–75.
- [M1] Macbeath, A.M.: On a curve of genus 7, *Proc. London Math. Soc.* **15** (1965), 527–542.
- [M2] Mahler, K.: *Lectures on Transcendental Numbers*. Berlin: Springer, 1976; Lecture Notes in Math. **546**.
- [M3] Mason, R.C.: *Diophantine Equations over Function Fields*, London Math. Soc. Lecture Notes Series #96, Cambridge Univ. Press, 1984. See also pages 149–157 in Springer LNM **1068** (1984) [=proceedings of Journées Arithmétiques 1983, Noordwijkerhout].
- [O] Oesterlé, J.: Nouvelles approches du “théorème” de Fermat, *Sém. Bourbaki* 2/88, exposé #694.
- [P] Pila, J.: Geometric postulation of a smooth function and the number of rational points, *Duke Math. J.* **63** (1991), 449–463.
- [PV] Payne, G., Vaserstein, L.N.: Sums of three cubes. Pages 443–454 in *The Arithmetic of Function Fields*, de Gruyter, 1992.
- [S] Stahlke, C.: Algebraic curves over \mathbb{Q} with many rational points and minimal automorphism group. *International Math. Research Notices* 1997 #1, 1–4.
- [T] Takeuchi, K.: Commensurability classes of arithmetic triangle groups, *J. Fac. Sci. Univ. Tokyo* **24** (1977), 201–212.
- [W1] Weil, A.: Abstract versus classical algebraic geometry. Pages 550–558 of *Proceedings of the International Congress of Mathematicians, 1954, Amsterdam, Vol. III*.
- [W2] Wildanger, K.: *Über das Lösen von Einheiten- und Indexformgleichungen in algebraischen Zahlkörpern mit einer Anwendung auf die Bestimmung aller ganzen Punkte einer Mordellschen Kurve*. Ph.D. Thesis, TU Berlin, Berlin, 1997.

Coverings of Curves of Genus 2

E. Victor Flynn

Department of Mathematical Sciences, University of Liverpool
Liverpool L69 3BX, United Kingdom
evflynn@liverpool.ac.uk

Abstract. We shall discuss the idea of finding all rational points on a curve \mathcal{C} by first finding an associated collection of curves whose rational points cover those of \mathcal{C} . This classical technique has recently been given a new lease of life by being combined with descent techniques on Jacobians of curves, Chabauty techniques, and the increased power of software to perform algebraic number theory. We shall survey recent applications during the last 5 years which have used Chabauty techniques and covering collections of curves of genus 2 obtained from pullbacks along isogenies on their Jacobians.

1 Introduction

We consider a general curve of genus 2 defined over a number field K

$$\mathcal{C} : Y^2 = F(X) = f_6X^6 + f_5X^5 + \dots + f_0 = F_1(X)\dots F_k(X), \quad (1)$$

where $F_1(X), \dots, F_k(X)$ are the irreducible factors of $F(X)$ over K ; we assume that $F(X)$ has no repeated roots and that $f_6 \neq 0$ or $f_5 \neq 0$. The intention is to describe, in a way accessible to a non-specialist, recent developments in Chabauty and covering techniques. These techniques all use essentially the same idea; we first find an Abelian variety A which maps to \mathcal{J} , the Jacobian of \mathcal{C} , under an isogeny ϕ . The pullbacks under ϕ of a suitably chosen set of embeddings of \mathcal{C} in \mathcal{J} , give a collection of curves lying on A whose rational points cover those of \mathcal{C} . Despite this rather geometric description, the mechanics of this, in the cases we shall consider, do not in fact require any difficult geometry. Provided the reader is prepared to take on faith a few standard results, the equations for the covering collections of curves can be obtained directly from that of \mathcal{C} .

In Section 2, we shall define the Jacobian of a curve of genus 2, and outline a few standard techniques for trying to find its rank. In Section 3, we describe Chabauty's Theorem and, in particular, how it can be applied to the problem of finding the K -rational points on a curve of genus 2 defined over K ; similar ideas can also be applied to an elliptic curve \mathcal{E} defined over a number field K , when one wants to find all points in $\mathcal{E}(K)$ subject to some arithmetic condition, such as the \mathbb{Q} -rationality of the x -coordinate. In Sections 4, 5, we describe the covering collections associated to various choices of isogeny, and give applications. Finally, in Section 6, we compare these techniques with a more classical approach using resultants. We shall try, in all sections, to provide sufficient detail that the non-specialist reader gains an impression of the techniques and difficulties involved.

We shall have in mind several motivating examples. The first of these concerns cycles of quadratic polynomials. Given a quadratic polynomial $az^2 + bz + c$, with $a, b, c \in \mathbb{Q}$ and $a \neq 0$, we say that $z \in \mathbb{Q}$ is a point of *exact period N* if $g^N(z) = z$ and $g^n(z) \neq z$ for all $n < N$. For example, $z = 0$ is a point of exact period 2 for $z^2 - 1$. It is easy to find such examples for $N = 1, 2, 3$, and it was shown in [19] that none exist for $N = 4$. We shall later summarise the proof in [14] of the fact that none exist for $N = 5$ also. It remains an unsolved problem whether any examples exist for $N \geq 6$. Applying a linear transformation on z , we can assume that our quadratic is monic and has no linear term; that is, it is of the form $g(z) = z^2 + c$ for some $c \in \mathbb{Q}$. Suppose that z is a point of exact period 5; then z, c must satisfy the curve $(g^5(z) - z)/(g(z) - z) = 0$. This curve in z, c is of degree 30 and genus 14, but it has a quotient \mathcal{C}_1 of genus 2, derived in [14], given by

$$\mathcal{C}_1 : Y^2 = X^6 + 8X^5 + 22X^4 + 22X^3 + 5X^2 + 6X + 1. \quad (2)$$

There are six obvious points $\infty^\pm, (0, \pm 1), (-3, \pm 1)$, where ∞^+, ∞^- denote the points on the non-singular curve that lie over the singular point at infinity on \mathcal{C} (for any curve (1) with $f_6 \neq 0$ both ∞^+ and ∞^- are in $\mathcal{C}(K)$ when $f_6 \in (K^*)^2$). These six points do not have preimages corresponding to $z, c \in \mathbb{Q}$ with z a point of exact period 5, and so the following Lemma gives a way of resolving the case $N = 5$.

Lemma 1. *Let \mathcal{C}_1 be as in (2). If $\mathcal{C}_1(\mathbb{Q}) = \{\infty^\pm, (0, \pm 1), (-3, \pm 1)\}$ then there is no quadratic polynomial in $\mathbb{Q}[z]$ with a rational point of exact period 5.*

Another application is to the equation

$$a^2 + b^2 = c^2, \quad a^3 + b^3 + c^3 = d^3, \quad a, b, c, d \in \mathbb{Z}. \quad (3)$$

There are the obvious solutions $(3, 4, 5, 6), (4, 3, 5, 6), (1, 0, -1, 0), (0, 1, -1, 0)$, and we would like to show that these are all of them up to scalar multiplication. The first solution, the so-called “Nuptial Number of Plato”, is thought (see [29]) to be mentioned indirectly in Plato’s *Republic*, as being a special relationship between the 3-4-5 triangle (viewed at the time as the marriage triangle between the “male” number 3 and “female” number 4) and the first perfect number 6. It is shown in [29] that there are no other solutions, using 15 pages of lengthy but elementary resultant and congruence arguments. We shall give a different proof here, using the ideas of the next two sections. For the moment, we merely observe that, on dividing through by c , we get equations in the three affine variables $A = a/c, B = b/c, D = d/c$. Furthermore, the solutions to $A^2 + B^2 = 1$ can be parametrised as $A = (1 - s^2)/(1 + s^2), B = 2s/(1 + s^2)$. We substitute these into $A^3 + B^3 + 1 = D^3$, multiply though by $(1 + s^2)^3$, and replace D by $t = D(1 + s^2)$ to give the curve

$$t^3 = 6s^4 + 8s^3 + 2, \quad (4)$$

which is a plane quartic with a double point at $s = -1, t = 0$, and no other singularities, and so is of genus 2. Using a standard trick (see p.4 of [7]) which

involves mapping the double point to $(0,0)$ and then completing a square, we can birationally change variable to $X = t/(1+s), Y = 12s - 4 - t^3/(1+s)^3$, which gives the equation $Y^2 = X^6 + 32X^3 - 32$, with our four known solutions to (3) corresponding to $\infty^\pm, (1, \pm 1)$.

Lemma 2. *Let $\mathcal{C}_2 : Y^2 = X^6 + 32X^3 - 32$. If $\mathcal{C}_2(\mathbb{Q}) = \{\infty^\pm, (1, \pm 1)\}$ then the only solutions to (3) are $(3, 4, 5, 6), (4, 3, 5, 6), (1, 0, -1, 0), (0, 1, -1, 0)$ up to scalar multiples.*

Also of historical interest is Problem 17 of book VI of the Arabic manuscript of *Arithmetica* [22]. Diophantus poses a problem equivalent to finding a non-trivial rational point on the genus 2 curve

$$\mathcal{C}_3 : Y^2 = X^6 + X^2 + 1. \quad (5)$$

The related problem of finding all rational points has recently been solved by Wetherell [28], who showed that $\mathcal{C}_3(\mathbb{Q}) = \{\infty^\pm, (0, \pm 1), (\pm 1/2, \pm 9/8)\}$ using Jacobians and covering techniques. We shall later give a sketch of the proof. This appears to be the only curve considered by Diophantus which has genus > 1 .

Another application, close to the heart of anyone who wants to construct exercises for a calculus class, is that of \mathbb{Q} -derived polynomials; that is, polynomials defined over \mathbb{Q} , with all derivatives having all of their roots in \mathbb{Q} . An example is $f(x) = x(x-1)(x-8/3)$, $f'(x) = (3x-4/3)(x-2)$, $f''(x) = 6x - 22/3$. We say that a polynomial is of type p_{m_1, \dots, m_r} if it has r distinct roots, and each m_i is the multiplicity of the i -th root. Two \mathbb{Q} -derived polynomials $q_1(x)$ and $q_2(x)$ are *equivalent* if $q_2(x) = r q_1(sx+t)$, for some constants $r, s, t \in \mathbb{Q}$, with $r, s \neq 0$. The problem of classifying all \mathbb{Q} -derived polynomials has been reduced in [5] to showing the following two conjectures.

Conjecture 1. *No polynomial of type $p_{1,1,1,1}$ is \mathbb{Q} -derived.*

Conjecture 2. *No polynomial of type $p_{3,1,1}$ is \mathbb{Q} -derived.*

Indeed, the following is shown in [5].

Theorem 1. *If Conjectures 1 and 2 are true then all \mathbb{Q} -derived polynomials are equivalent to one of*

$$x^n, x^{n-1}(x-1), x(x-1)\left(x - \frac{v(v-2)}{v^2-1}\right), x^2(x-1)\left(x - \frac{9(2w+z-12)(w+2)}{(z-w-18)(8w+z)}\right),$$

for some $n \in \mathbb{Z}^+$, $v \in \mathbb{Q}$, $(w, z) \in \mathcal{E}_0(\mathbb{Q})$, where $\mathcal{E}_0 : z^2 = w(w-6)(w+18)$ is an elliptic curve of rank 1.

For Conjecture 2, we let $q(x)$ be a \mathbb{Q} -derived polynomial of type $p_{3,1,1}$, which we may take to be in the form $q(x) = x^3(x-1)(x-a)$, for some $a \in \mathbb{Q}$ with $a \neq 0, 1$. The discriminants of the quadratics $q'''(x)$, $q''(x)/x$ and $q'(x)/x^2$, must

all be rational squares, and so must be their product. This implies that a satisfies $(4a^2 - 7a + 4)(9a^2 - 12a + 9)(4a^2 - 2a + 4) = b^2$, for some $b \in \mathbb{Q}$. Using the transformation $a = (X - 3)/(X + 3)$, $b = 6Y/(X + 3)^3$ gives the genus 2 curve

$$\mathcal{C}_4 : Y^2 = (X^2 + 15)(X^2 + 45)(X^2 + 135). \quad (6)$$

The obvious points $\infty^\pm, (\pm 3, \pm 432)$ correspond to the illegal values $a = 0, 1, \infty$, and so it is sufficient to show there are no others.

Lemma 3. *Let \mathcal{C}_4 be as in (6). If $\mathcal{C}_4(\mathbb{Q}) = \{\infty^\pm, (\pm 3, \pm 432)\}$ then Conjecture 2 is true.*

In Section 4, we shall sketch the proof in [16] that this is indeed all of $\mathcal{C}_4(\mathbb{Q})$, and so now only Conjecture 1 (a surface) remains unsolved.

A very recent result has been the solution in [17] of the “Serre curve”

$$\mathcal{D} : x^4 + y^4 = 17. \quad (7)$$

Serre asks (p.67 of [21]) whether $(x, y) = (\pm 1, \pm 2), (\pm 2, \pm 1)$ are the only $x, y \in \mathbb{Q}$ satisfying (7). This curve is the only Fermat quartic of the type $x^4 + y^4 = c$, with $c \leq 81$, which cannot trivially be solved by local methods or by a map onto an elliptic curve of rank 0. It has gained some notoriety as being resistant to various methods of attack, but has finally succumbed to the general method we shall briefly mention in Section 5.

The work of Bruin develops related ideas, which have been applied with great success to equations of the type $x^p + y^q = z^r$. We shall mention two of these, and give an indication of the approach used.

2 Preliminary Definitions

At the risk of insulting the reader’s intelligence, we shall briefly summarise a few standard facts about elliptic curves. Consider the elliptic curve defined over K

$$\mathcal{E} : y^2 = G(x) = g_3x^3 + g_2x^2 + g_1x + g_0 = G_1(x) \dots G_k(x), \quad (8)$$

where $G(x)$ has no repeated roots, $g_3 \neq 0$, and $G_1(x), \dots, G_k(x)$ are the irreducible factors of $G(x)$ over K . Let ∞ denote the point at infinity, which we take to be the identity in the group $\mathcal{E}(K)$ of K -rational points on \mathcal{E} . The rules $-(x, y) = (x, -y)$ and $P+Q+R = \infty \iff P, Q, R$ are collinear, are sufficient to compute the group law on $\mathcal{E}(K)$, and the points of order 2 are of the form $(x, 0)$, where $x \in K$ is a root of $G(x)$. The Mordell-Weil Theorem gives that $\mathcal{E}(K)$ is isomorphic to $\mathcal{E}(K)_{\text{tor}} \times \mathbb{Z}^r$, where $\mathcal{E}(K)_{\text{tor}}$ is the subgroup of $\mathcal{E}(K)$ consisting of points of finite order, and r is the rank of $\mathcal{E}(K)$. The finite group $\mathcal{E}(K)_{\text{tor}}$ is normally found by using reduction maps modulo primes of good reduction. For each $i \in \{1, \dots, k\}$ let α_i be a root of $G_i(x)$ and let $L_i = K(\alpha_i)$.

Define the homomorphism

$$\mu : \mathcal{E}(K) \rightarrow L_1^*/(L_1^*)^2 \times \dots \times L_k^*/(L_k^*)^2, \quad (x, y) \mapsto [g_3(x-\alpha_1), \dots, g_3(x-\alpha_k)], \quad (9)$$

which has kernel $2\mathcal{E}(K)$. Here, $g_3(x - \alpha_j)$ is taken to be 1 when $(x, y) = \infty$, and $\prod_{i \neq j} (x - \alpha_i)$ when $x = \alpha_j$. If we let $S = \{2, p_1, \dots, p_m\}$, where p_1, \dots, p_m are the rational primes of bad reduction, then the image of q is contained inside the finite group M , consisting of those $[d_1, \dots, d_k]$ such that all of the field extensions $L_1(\sqrt{d_1}) : L_1, \dots, L_k(\sqrt{d_k}) : L_k$ are unramified outside of primes lying over primes of S . Once M is determined, one eliminates members of M as potential members of $\text{im}(q)$ by local (congruence) arguments. What remains is the 2 -Selmer group, and one hopes that this is enough to determine the 2-rank of $\text{im}(q)$, and hence that of $\mathcal{E}(K)/2\mathcal{E}(K)$. If so, then one will have performed a successful complete 2-descent. On subtracting the 2-rank of $\mathcal{E}(K)_{\text{tor}}/2\mathcal{E}(K)_{\text{tor}}$, the remainder is the rank of $\mathcal{E}(K)$. A benefit of recent developments in algebraic number theory software, such as PARI/GP [1] and KASH [10], is that the above approach has become possible for elliptic curves defined over increasingly complicated number fields. Some of the methodology is described in [11],[20],[23],[24]; see also the program [4]. We mention here two such computations in the literature ([15], [16], respectively) which will be relevant to later sections.

Example 1. Let α satisfy $\alpha^3 + \alpha + 1 = 0$, and define over $\mathbb{Q}(\alpha)$ the elliptic curves $\mathcal{E}_1 : y^2 = x(x^2 + \alpha x + (\alpha^2 + 1))$ and $\mathcal{E}_2 : y^2 = -\alpha x(x^2 + \alpha x + (\alpha^2 + 1))$. Then $\mathcal{E}_1(\mathbb{Q}(\alpha))$ has rank 1, with generators given by $(0, 0), (-\alpha, 1)$, where $(0, 0)$ is of order 2 and $(-\alpha, 1)$ is of infinite order. Also, $\mathcal{E}_2(\mathbb{Q}(\alpha))$ has rank 0 and consists only of ∞ and $(0, 0)$.

Example 2. Let $\beta = \sqrt{-15}$ and let $\mathcal{E}_3 : y^2 = 6(54 + 6\beta)(-45x^2 + 1)(\beta x + 1)$ and $\mathcal{E}_4 : y^2 = 6(9 + \beta)(-45x^2 + 1)(\beta x + 1)$. Then $\mathcal{E}_3(\mathbb{Q}(\beta))$ has rank 1 and is generated by the 2-torsion point $(-1/\beta, 0)$ and the point $(1/6 + \beta/30, 24)$ of infinite order. Similarly, $\mathcal{E}_4(\mathbb{Q}(\beta))$ has rank 1 and is generated by the 2-torsion point $(-1/\beta, 0)$ and the point $(-1/6 + \beta/30, 9 + \beta)$ of infinite order.

Given a curve \mathcal{C} of genus 2, as in (1) with $f_6 \neq 0$, we use ∞^+, ∞^- as described after (2). When $f_6 = 0$ (and so $f_5 \neq 0$), we let ∞ denote the point at infinity, which is always in $\mathcal{C}(K)$. Following Chapter 1 of [7], any member of $\mathcal{J}(K)$, the K -rational points on the Jacobian, may be represented by a divisor of the form $P_1 + P_2 - \infty^+ - \infty^-$, where P_1, P_2 are points on \mathcal{C} and either P_1, P_2 are both K -rational or P_1, P_2 are quadratic over K and conjugate. We shall abbreviate such a divisor by: $\{P_1, P_2\}$. This representation gives a 1-1 correspondence with members of $\mathcal{J}(K)$, except that everything of the form $\{(X, Y), (X, -Y)\}$ must be identified into a single equivalence class \mathcal{O} , which serves as the group identity in $\mathcal{J}(K)$. Note that $-\{(x_1, y_1), (x_2, y_2)\} = \{(x_1, -y_1), (x_2, -y_2)\}$; furthermore $\{P_1, P_2\} + \{Q_1, Q_2\} + \{R_1, R_2\} = \mathcal{O}$ if and only if there exists $Y(X)$ of degree ≤ 3 such that $Y = Y(X)$ meets \mathcal{C} at $P_1, P_2, Q_1, Q_2, R_1, R_2$. These two rules are sufficient for computing the group law on $\mathcal{J}(K)$. Clearly, an element of order 2 in $\mathcal{J}(K)$ is given by $\{(X_1, 0), (X_2, 0)\}$, where X_1, X_2 are the roots of quadratic $Q(X)$ defined over K , satisfying $Q(X)|F(X)$. The Mordell-Weil Theorem gives that $\mathcal{J}(K)$ is isomorphic to $\mathcal{J}(K)_{\text{tor}} \times \mathbb{Z}^r$, where $\mathcal{J}(K)_{\text{tor}}$ is the subgroup of $\mathcal{J}(K)$ consisting of points of finite order, and r is the rank of $\mathcal{J}(K)$. The finite group $\mathcal{J}(K)_{\text{tor}}$ is normally found by using reduction maps modulo

primes of good reduction. For each $i \in \{1, \dots, k\}$ let α_i be a root of $F_i(x)$ and let $L_i = K(\alpha_i)$. When $f_6 \neq 0$, we define the homomorphism

$$\begin{aligned} \mu : \mathcal{J}(K) &\rightarrow \left(L_1^*/(L_1^*)^2 \times \dots \times L_k^*/(L_k^*)^2 \right) / \sim, \\ &: \{(X_1, Y_1), (X_2, Y_2)\} \mapsto [(X_1 - \alpha_1)(X_2 - \alpha_2), \dots, (X_1 - \alpha_k)(X_2 - \alpha_k)], \end{aligned} \quad (10)$$

where the equivalence relation \sim is defined by

$$[a_1, \dots, a_k] \sim [b_1, \dots, b_k] \iff a_1 = wb_1, \dots, a_k = wb_k, \text{ for some } w \in K^*. \quad (11)$$

The interpretations of $X_i - \alpha_j$ in special cases where (X_i, Y_i) is a point at infinity, or $X_i = \alpha_j$, are as described immediately after (9). Either $2\mathcal{J}(K)$ is the kernel of q or it has index 2 in the kernel of q (see [14]). The image of q is contained inside a finite group M , which is as described above for elliptic curves. Once M is determined, one proceeds in a similar manner to the complete 2-descent for elliptic curves described above, and hopes to find [26] the 2-rank of $\mathcal{J}(K)/2\mathcal{J}(K)$. There is some extra finesse here in determining whether or not the kernel of q is $2\mathcal{J}(K)$, and in the interpretation of the local information; there is also the potential for difficult computations in number fields of higher degree over the ground field than for elliptic curves. When $f_6 = 0$, the relation \sim can be removed, and the mechanics become more similar to that of complete 2-descent on an elliptic curve. As with elliptic curves, the final step is to subtract the 2-rank of $\mathcal{J}(K)_{\text{tor}}/2\mathcal{J}(K)_{\text{tor}}$ from that of $\mathcal{J}(K)/2\mathcal{J}(K)$ to obtain the rank of $\mathcal{J}(K)$. Recent developments in canonical heights and infinite descent ([13], [25], [27]) also allow actual generators for $\mathcal{J}(K)$ to be computed in many cases. We mention here three ranks computed in the literature ([14], [28], [16], respectively), which we shall require later. Only the first of these is a genuine genus 2 computation, the other three ranks being computable via maps to elliptic curves.

Example 3. Let $\mathcal{C}_1 : Y^2 = X^6 + 8X^5 + 22X^4 + 22X^3 + 5X^2 + 6X + 1$, as in (2), with Jacobian \mathcal{J}_1 . Then $\mathcal{J}_1(\mathbb{Q})_{\text{tor}} = \{\mathcal{O}\}$; the rank of $\mathcal{J}_1(\mathbb{Q})$ is 1, and it is generated by $\{\infty^+, \infty^+\}$.

Example 4. Let $\mathcal{C}_2 : Y^2 = X^6 + 32X^3 - 32$, as in Lemma 2, with Jacobian \mathcal{J}_2 . Then $\mathcal{J}_2(\mathbb{Q})_{\text{tor}} = \{\mathcal{O}, \{\infty^+, \infty^+\}, \{\infty^-, \infty^-\}\}$; the rank of $\mathcal{J}_2(\mathbb{Q})$ is 1, and it is generated by $\mathcal{J}_2(\mathbb{Q})_{\text{tor}}$ and $\{(1, 1), \infty^+\}$.

Example 5. Let $\mathcal{C}_3 : Y^2 = X^6 + X^2 + 1$, with Jacobian \mathcal{J}_3 . Then $\mathcal{J}_3(\mathbb{Q})_{\text{tor}} = \{\mathcal{O}\}$; the rank of $\mathcal{J}_3(\mathbb{Q})$ is 2, and it is generated by $\{(0, 1), (0, 1)\}$ and $\{(0, 1), \infty^+\}$.

Example 6. Let $\mathcal{C}_4 : Y^2 = F_1(X)F_2(X)F_3(X)$, with Jacobian \mathcal{J}_4 , where:

$$F_1(X) = X^2 + 15, \quad F_2(X) = X^2 + 45, \quad F_3(X) = X^2 + 135,$$

and let α_i, β_i be the roots of $G_i(X)$ for $1 \leq i \leq 3$. Then

$\mathcal{J}_4(\mathbb{Q})_{\text{tor}} = \{\mathcal{O}, \{(\alpha_1, 0), (\beta_1, 0)\}, \{(\alpha_2, 0), (\beta_2, 0)\}, \{(\alpha_3, 0), (\beta_3, 0)\}\}$; the rank of $\mathcal{J}_4(\mathbb{Q})$ is 2, and it is generated by the 2-torsion above, together with $\{\infty^+, \infty^+\}$ and $\{(3, 432), \infty^+\}$.

3 Chabauty's Theorem

Let \mathcal{E} be an elliptic curve, as in (8), defined over a number field $K = \mathbb{Q}(\alpha)$ of degree d . We shall consider the problem of trying to find all

$$(x, y) \in \mathcal{E}(\mathbb{Q}(\alpha)) \text{ with } x \in \mathbb{Q}. \quad (12)$$

Imitating Chapter IV of [24] (with the difference that our equations include g_3 , the coefficient of x^3), we introduce the variables $s = -x/y, w = -1/y$. Then $w = g_3s^3 + g_2s^2w + g_1sw^2 + g_0w^3$, and recursive substitution gives $w = w(s)$, a power series in the local parameter s , with initial term g_3s^3 . Then $1/x = w(s)/s$ is a power series

$$\frac{1}{x}(s) = g_3(s^2 + g_2s^4 + (g_1g_3 + g_2^2)s^6 + O(s^8)) \in \mathbb{Z}[g_0, g_1, g_2, g_3][[s]]. \quad (13)$$

If (x_0, y_0) is another point on \mathcal{E} , then the x -coordinate of $(x_0, y_0) + (x, y)$ is a power series

$$\begin{aligned} x\text{-coord of } ((x_0, y_0) + (x, y)) &= x_0 + 2y_0s + (3g_3x_0^2 + 2g_2x_0 + g_1)s^2 + O(s^3) \\ &\in \mathbb{Z}[g_0, g_1, g_2, g_3, x_0, y_0][[s]]. \end{aligned} \quad (14)$$

If $(s, w(s)), (t, w(t))$ are two points in s - w coordinates then the s -coordinate of the sum can be written as $\mathcal{F}(s, t) \in \mathbb{Z}[g_0, g_1, g_2, g_3][[s, t]]$, the *formal group*. There are then power series

$$\log(t) = t + \frac{1}{3}g_2t^3 + \frac{1}{5}(g_2^2 + 2g_1g_3)t^5 + O(t^7) \in \mathbb{Q}[g_0, g_1, g_2, g_3][[t]], \quad (15)$$

$$\exp(t) = t - \frac{1}{3}g_2t^3 + \frac{1}{15}(2g_2^2 - 6g_1g_3)t^5 + O(t^7) \in \mathbb{Q}[g_0, g_1, g_2, g_3][[t]], \quad (16)$$

satisfying $\log(\mathcal{F}(s, t)) = \log(s) + \log(t)$, $\mathcal{F}(\exp(s), \exp(t)) = \exp(s + t)$. In either power series, the denominator of the coefficient of t^k divides $k!$.

We now suppose that the rank r of $\mathcal{E}(\mathbb{Q}(\alpha))$ is less than $d = [\mathbb{Q}(\alpha) : \mathbb{Q}]$, and that we have found generators for $\mathcal{E}(\mathbb{Q}(\alpha))$:

$$\mathcal{E}(\mathbb{Q}(\alpha)) = \langle \mathcal{E}(\mathbb{Q}(\alpha))_{\text{tor}}, P_1, \dots, P_r \rangle. \quad (17)$$

Suppose that p is an odd prime such that $|\alpha|_p = 1$, $\mathbb{Q}(\alpha)$ is unramified at p , \mathcal{E} has good reduction at p , $[\mathbb{Q}_p(\alpha) : \mathbb{Q}_p] = [\mathbb{Q}(\alpha) : \mathbb{Q}] = d$, and $|g_i|_p \leq 1$, for $i = 1, \dots, 3$. These restrictions on p (which cannot be satisfied for some choices of α) are only for the sake of simplifying the exposition. Let $\tilde{\alpha}, \tilde{\mathcal{E}}, \tilde{P}_1, \dots, \tilde{P}_r$ represent, respectively, the reductions mod p of $\alpha, \mathcal{E}, P_1, \dots, P_r$. Further define $m_i, Q_i, x_i, y_i, s^{(i)}$ by

$$m_i = \text{order of } \tilde{P}_i \text{ in } \tilde{\mathcal{E}}(\mathbb{F}_p(\tilde{\alpha})), \quad Q_i = m_i P_i = (x_i, y_i), \quad s^{(i)} = -x_i/y_i, \quad (18)$$

so that each $Q_i \in \mathcal{E}(\mathbb{Q}(\alpha))$ is in the kernel of the reduction map from $\mathcal{E}(\mathbb{Q}(\alpha))$ to $\tilde{\mathcal{E}}(\mathbb{F}_p(\tilde{\alpha}))$, giving $|s^{(i)}|_p \leq p^{-1}$. Now, let \mathcal{S} be a set (which must be finite) of

representatives of $\mathcal{E}(\mathbb{Q}(\alpha))$ modulo $\langle Q_1, \dots, Q_r \rangle$, so that every $P \in \mathcal{E}(\mathbb{Q}(\alpha))$ can be written uniquely in the form

$$P = S + n_1 Q_1 + \dots + n_r Q_r, \quad (19)$$

for some $S \in \mathcal{S}$ and $n_1, \dots, n_r \in \mathbb{Z}$. We can now express the s -coordinate of $n_1 Q_1 + \dots + n_r Q_r$, using (15), (16), as: $\exp(n_1 \log(s^{(1)}) + \dots + n_r \log(s^{(r)}))$, which is a power series in n_1, \dots, n_r . Substituting this power series for s in (13) when $S = \infty$, and in (14) when $S = (x_0, y_0) \neq \infty$ gives

$$\theta_S(n_1, \dots, n_r) = x_S(S + n_1 Q_1 + \dots + n_r Q_r) \in \mathbb{Z}_p[\alpha][[n_1, \dots, n_r]], \quad (20)$$

where x_S means x -coordinate, when $S \neq \infty$, and $1/x$ -coordinate when $S = \infty$. It is clear, from the standard estimate $|k!|_p \geq p^{-(k-1)/(p-1)}$, that the coefficient of $n_1^{k_1} \dots n_r^{k_r}$ is in $\mathbb{Z}_p[\alpha]$, and converges to 0 as $k_1 + \dots + k_r \rightarrow \infty$. Splitting θ_S into its components

$$\theta_S = \theta_S^{(0)} + \theta_S^{(1)} \alpha + \dots + \theta_S^{(d-1)} \alpha^{d-1}, \text{ each } \theta_S^{(i)}(n_1, \dots, n_r) \in \mathbb{Z}_p[[n_1, \dots, n_r]], \quad (21)$$

we obtain power series satisfying

$$(x\text{-coord of } P) \in \mathbb{Q} \Rightarrow \theta_S^{(1)} = \dots = \theta_S^{(d-1)} = 0. \quad (22)$$

We now make use of the following theorem (p.62 of [6]).

Theorem 2. (Strassmann). Let $\theta(X) = c_0 + c_1 X + \dots \in \mathbb{Z}_p[[X]]$ satisfy $c_j \rightarrow 0$ in \mathbb{Z}_p . Define ℓ uniquely by: $|c_\ell|_p \geq |c_j|_p$ for all $j \geq 0$, and $|c_\ell|_p > |c_j|_p$ for all $j > \ell$. Then there are at most ℓ values of $x \in \mathbb{Z}_p$ such that $\theta(x) = 0$.

When r , the rank of $\mathcal{E}(\mathbb{Q}(\alpha))$, is 1 (as will be the case in the following examples), and $d = [\mathbb{Q}(\alpha) : \mathbb{Q}] > 1$, then we can apply Strassmann's Theorem to bound, for example, the number of roots of $\theta_S^{(1)}(n_1)$. In view of (22), summing these bounds over all $S \in \mathcal{S}$ gives an upper bound on the total number of (x, y) satisfying (12), which we hope to be the number of known such (x, y) . When $r > 1$ and $r < d$, we can in principle try to perform repeated applications if the Weierstrass Preparation Theorem (see p.108 of [6]) and resultant computations to derive univariate power series from d power series in r variables.

Example 7. Let $\alpha, \mathcal{E}_1, \mathcal{E}_2$ be as in Example 1. Then the only $(x, y) \in \mathcal{E}_1(\mathbb{Q}(\alpha))$ with $x \in \mathbb{Q}$ are $\infty, (0, 0), \pm(1/4, 1/8 - \alpha/2 + \alpha^2/4)$. The only $(x, y) \in \mathcal{E}_2(\mathbb{Q}(\alpha))$ with $x \in \mathbb{Q}$ are $\infty, (0, 0)$.

Proof (see [15] for details): The result on $\mathcal{E}_2(\mathbb{Q}(\alpha))$ follows immediately from Example 1, since the rank is 0, and $\infty, (0, 0)$ are the only members of $\mathcal{E}_2(\mathbb{Q}(\alpha))$.

For $\mathcal{E}_1(\mathbb{Q}(\alpha))$, let $P_1 = (-\alpha, 1)$, $p = 5$, $m_1 = 28$; then $28P_1$ is in the kernel of reduction mod 5, but it is more efficient to take $Q_1 = 14P_1 + (0, 0)$, which is also in the kernel of reduction mod 5. Let

$$\mathcal{S} = \{kP_1 : -6 \leq k \leq 7\} \cup \{(0, 0) + kP_1 : -6 \leq k \leq 7\}, \quad (23)$$

so that any $P \in \mathcal{E}_1(\mathbb{Q}(\alpha))$ can be written as $S + n_1 Q_1$, for some $S \in \mathcal{S}$, $n_1 \in \mathbb{Z}$. Let us first consider $S = -2P_1 = (1/4, 1/8 - \alpha/2 + \alpha^2/4)$. Applying (15), (16), gives the s -coordinate of $n_1 Q_1$ as:

$$\exp(n_1 \log(s\text{-coordinate of } Q_1)) \equiv 5(21 + 15\alpha + 21\alpha^2)n_1 \pmod{5^3}. \quad (24)$$

Replacing (x_0, y_0) by $(1/4, 1/8 - \alpha/2 + \alpha^2/4)$ and s by (24) in (14) gives the x -coordinate of $-2P_1 + n_1 Q_1$ as:

$$\theta_{-2P_1}(n_1) \equiv 94 + 5(17\alpha + 9\alpha^2)n_1 + 5^2(2 + \alpha + \alpha^2)n_1^2 \pmod{5^3}. \quad (25)$$

We may consider either $\theta_{-2P_1}^{(1)}$ or $\theta_{-2P_1}^{(2)}$, due to the fact that the rank of $\mathcal{E}(\mathbb{Q}(\alpha))$ is two less than $[\mathbb{Q}(\alpha) : \mathbb{Q}]$. Taking $\theta_{-2P_1}^{(2)}(n_1) \equiv 5 \cdot 9 \cdot n_1 + 5^2 \cdot n_1^2 \pmod{5^3}$, and applying Strassmann's Theorem, gives that there is at most one root; but we know that $n_1 = 0$ is a root, since $-2P_1 + 0 \cdot Q_1$ has x -coordinate $= 1/4 \in \mathbb{Q}$. Hence $n_1 = 0$ is the only solution. Similarly, for $S = \infty, (0, 0), 2P_1$ we can show that $n_1 = 0$ is the value of n_1 for which $S + n_1 Q_1$ can have \mathbb{Q} -rational x -coordinate. For the remaining ten values of $S \in \mathcal{S}$, we find that $\theta_S^{(2)}(n_1)$ has constant term of 5-adic norm strictly greater than all subsequent coefficients; hence there are no roots in these cases. In summary, we have shown that $\infty, (0, 0), \pm P_1$ are the only members of $\mathcal{E}(\mathbb{Q}(\alpha))$ with \mathbb{Q} -rational x -coordinate, as required. \square

A similar argument (see [16]), working mod 11^3 , shows the following.

Example 8. Let $\beta, \mathcal{E}_3, \mathcal{E}_4$ be as in Example 2. Then the only $(x, y) \in \mathcal{E}_3(\mathbb{Q}(\beta))$ with $x \in \mathbb{Q}$ are $\infty, \pm(-1/3, 12 + 12\beta), \pm(1/9, 12 + 4\beta/3)$. Similarly, the only $(x, y) \in \mathcal{E}_4(\mathbb{Q}(\beta))$ with $x \in \mathbb{Q}$ are $\infty, \pm(1/3, 12 - 4\beta), \pm(-1/9, 16/3)$.

Now, consider a curve (1) of genus 2; suppose that it is defined over \mathbb{Q} and that $\mathcal{J}(\mathbb{Q})$ has rank 1. Given $D = \{(X_1, Y_1), (X_2, Y_2)\} \in \mathcal{J}(\mathbb{Q})$, it is possible to describe a local parameter $\mathbf{s} = (s_1, s_2)$ given by

$$\begin{aligned} s_1 &= (\mathcal{G}_1(X_1, X_2)Y_1 - \mathcal{G}_1(X_2, X_1)Y_2)(X_1 - X_2)/(\mathcal{F}_0(X_1, X_2) - 2Y_1 Y_2)^2, \\ s_2 &= (\mathcal{G}_0(X_1, X_2)Y_1 - \mathcal{G}_0(X_2, X_1)Y_2)(X_1 - X_2)/(\mathcal{F}_0(X_1, X_2) - 2Y_1 Y_2)^2, \end{aligned} \quad (26)$$

where

$$\begin{aligned} \mathcal{F}_0(X_1, X_2) &= 2f_0 + f_1(X_1 + X_2) + 2f_2(X_1 X_2) + f_3(X_1 X_2)(X_1 + X_2) \\ &\quad + 2f_4(X_1 X_2)^2 + f_5(X_1 X_2)^2(X_1 + X_2) + 2f_6(X_1 X_2)^3, \\ \mathcal{G}_1(X_1, X_2) &= 2f_0(X_1 + X_2) + f_1 X_2(3X_1 + X_2) + 4f_2(X_1 X_2^2) \\ &\quad + f_3(X_1^2 X_2^2 + 3X_1 X_2^3) + f_4(2X_1^2 X_2^3 + 2X_1 X_2^4) \\ &\quad + f_5(3X_1^2 X_2^4 + X_1 X_2^5) + 4f_6(X_1^2 X_2^5), \\ \mathcal{G}_0(X_1, X_2) &= 4f_0 + f_1(X_1 + 3X_2) + f_2(2X_1 X_2 + 2X_2^2) + f_3(3X_1 X_2^2 + X_2^3) \\ &\quad + 4f_4(X_1 X_2^3) + f_5(X_1^2 X_2^3 + 3X_1 X_2^4) + f_6(2X_1^2 X_2^4 + 2X_1 X_2^5). \end{aligned}$$

The derivations of the above definitions are given in Chapter 7 of [7]. The reader can at least observe that s_1, s_2 will both be small when D is close to \mathcal{O} . It is sufficient, in what follows, to accept on faith that $\mathbf{s} = (s_1, s_2)$ performs the same role on $\mathcal{J}(\mathbb{Q})$ as $s = -x/y$ does on an elliptic curve. Let $D_0, D \in \mathcal{J}(\mathbb{Q})$,

with $\mathbf{s} = \mathbf{s}(D) = (s_1(D), s_2(D))$ being the local parameter for D , and let $D_0 + D = \{(X'_1, Y'_1), (X'_2, Y'_2)\}$. Then the group law on $\mathcal{J}(\mathbb{Q})$ can be applied to find $\psi_{D_0}^{(1)}(\mathbf{s}), \psi_{D_0}^{(2)}(\mathbf{s}), \psi_{D_0}^{(3)}(\mathbf{s})$, power series in \mathbf{s} , such that

$$(1 : X'_1 + X'_2 : X'_1 X'_2) = (\psi_{D_0}^{(1)}(\mathbf{s}) : \psi_{D_0}^{(2)}(\mathbf{s}) : \psi_{D_0}^{(3)}(\mathbf{s})), \quad (27)$$

where both sides should be viewed as projective triples. Associated to our local parameter is $\mathcal{F}(\mathbf{s}, \mathbf{t})$, the two-parameter formal group of $\mathcal{J}(\mathbb{Q})$, the formal logarithm $L = (L_1, L_2)$ and exponential map $E = (E_1, E_2)$, given by

$$\begin{aligned} L_1(\mathbf{s}) &= s_1 + \frac{1}{3}(-2f_4s_1^3 + f_1s_2^3) + \dots \\ L_2(\mathbf{s}) &= s_2 + \frac{1}{3}(-2f_2s_2^3 + f_5s_1^3) + \dots \end{aligned} \quad (28)$$

These satisfy $L(\mathcal{F}(\mathbf{s}, \mathbf{t})) = L(\mathbf{s}) + L(\mathbf{t})$ and $E(\mathbf{s} + \mathbf{t}) = \mathcal{F}(E(\mathbf{s}), E(\mathbf{t}))$. Now, suppose that $\mathcal{J}(\mathbb{Q}) = \langle J(\mathbb{Q})_{\text{tor}}, D_1 \rangle$, and let p be a prime of good reduction. Let $\tilde{\mathcal{J}}$ and \tilde{D}_1 represent, respectively, the reductions mod p of J and D_1 . Further define $m_1, E_1, \mathbf{s}^{(1)}$ by

$$m_1 = \text{order of } \tilde{D}_1 \text{ in } \tilde{\mathcal{J}}(\mathbb{F}_p), \quad E_1 = m_1 D_1, \quad \mathbf{s}^{(1)} = \mathbf{s}(D_1), \quad (29)$$

so that $E_1 \in \mathcal{J}(\mathbb{Q})$ is in the kernel of the reduction map from $\mathcal{J}(\mathbb{Q})$ to $\tilde{\mathcal{J}}(\mathbb{F}_p)$, giving $|s_1^{(1)}|_p, |s_2^{(1)}|_p \leq p^{-1}$. Now, let \mathcal{S} be a set (which must be finite) of representatives of $\mathcal{J}(\mathbb{Q})$ modulo $\langle E_1 \rangle$, so that every $D \in \mathcal{J}(\mathbb{Q})$ can be written uniquely in the form

$$D = S + n_1 E_1, \quad (30)$$

for some $S \in \mathcal{S}$ and $n_1 \in \mathbb{Z}$. Now express $\mathbf{s}(D)$, using (28), as: $\exp(n_1 \log(\mathbf{s}^{(1)}))$, which is a power series in n_1 . Substitute this power series for \mathbf{s} in (27) and take $D_0 = S$ to obtain

$$\theta_S^{(i)}(n_1) = \psi_S^{(i)}(\exp(n_1 \log(\mathbf{s}^{(1)}))) \in \mathbb{Z}_p[[n_1]], \quad \text{for } i = 1, 2, 3. \quad (31)$$

As with elliptic curves, the standard estimate $|k!|_p \geq p^{-(k-1)/(p-1)}$, can be used to show that the coefficient of n_1^k is in \mathbb{Z}_p , and converges to 0 as $k \rightarrow \infty$.

So far, what we have achieved is to find a finite set of triples of power series, namely $(\theta_S^{(1)}(n_1), \theta_S^{(2)}(n_1), \theta_S^{(3)}(n_1))$ for $S \in \mathcal{S}$, such that any $D \in \mathcal{J}(\mathbb{Q})$ has $(1 : X_1 + X_2 : X_1 X_2)$ equal to one of them. Now recall our original purpose, to find all of $\mathcal{C}(\mathbb{Q})$. The strategy is to embed the curve \mathcal{C} into its Jacobian; we shall choose the map $P \mapsto \{P, P\}$, for any $P \in \mathcal{C}(\mathbb{Q})$. This is not quite an injection, since any $(X, 0) \mapsto \mathcal{O}$; however, it is straightforward to find all \mathbb{Q} -rational roots of the sextic $F(X)$, and so all points $(X, 0) \in \mathcal{C}(\mathbb{Q})$. Therefore, we can set these aside and concentrate on $P = (X, Y)$ with $Y \neq 0$, where $P \mapsto \{P, P\}$ is injective. It is sufficient, then, to find all $D \in \mathcal{J}(\mathbb{Q})$ of the form $D = \{P, P\}$. Note that this implies $X_1 = X_2$, and so $(X_1 + X_2)^2 - 4X_1 X_2 = 0$, giving

$$\theta_S^{(2)}(n_1)^2 - 4\theta_S^{(1)}(n_1)\theta_S^{(3)}(n_1) = 0, \quad (32)$$

for some $S \in \mathcal{S}$ – namely the $S \in \mathcal{S}$ such that $D = S \bmod \langle E_1 \rangle$. Our strategy, then, is to compute the power series in (32) and use Strassmann’s Theorem to find an upper bound on the number of possible n_1 . Adding these bounds together gives an upper bound on the number of $(X, Y) \in \mathcal{C}(\mathbb{Q})$ with $Y \neq 0$, which we hope to be the same as the number of known points. We illustrate this with the following example from [14].

Example 9. Let \mathcal{C}_1 be as in Example 3. Then $\mathcal{C}_1(\mathbb{Q}) = \{\infty^\pm, (0, \pm 1), (-3, \pm 1)\}$.

Proof. We already know from Example 3 that $\mathcal{J}_1(\mathbb{Q})$ has no nontrivial torsion and has rank 1, with $\mathcal{J}_1(\mathbb{Q}) = \langle D_1 \rangle$, where $D_1 = \{\infty^+, \infty^-\}$. Let $p = 3$, which is a prime of good reduction, since the discriminant of the sextic is $2^{12} \cdot 3701$. Let $\tilde{D}_1 \in \tilde{\mathcal{J}}(\mathbb{F}_3)$ denote the reduction of $D_1 \bmod 3$. The following lists the first few multiples of D_1 and \tilde{D}_1 . In the table, which is reproduced from [14], $P_0 = (-2 + \frac{1}{3}\sqrt{33}, -\frac{17}{3} + \frac{10}{9}\sqrt{33})$ and $Q_0 = (-\frac{1}{2} + \frac{1}{6}\sqrt{-87}, \frac{22}{3} + \frac{5}{9}\sqrt{-87})$, and \overline{P}_0 and \overline{Q}_0 are their conjugates over \mathbb{Q} .

n	nD_1	$n\tilde{D}_1$
0	\mathcal{O}	\mathcal{O}
1	$\{\infty^+, \infty^+\}$	$\{\infty^+, \infty^+\}$
2	$\{(0, 1), (-3, 1)\}$	$\{(0, 1), (0, 1)\}$
3	$\{(0, -1), \infty^-\}$	$\{(0, -1), \infty^-\}$
4	$\{(0, -1), \infty^+\}$	$\{(0, -1), \infty^+\}$
5	$\{(-3, 1), \infty^-\}$	$\{(0, 1), \infty^-\}$
6	$\{(-3, 1), \infty^+\}$	$\{(0, 1), \infty^+\}$
7	$\{(0, -1), (0, -1)\}$	$\{(0, -1), (0, -1)\}$
8	$\{P, \overline{P}\}$	$\{\infty^-, \infty^-\}$
9	$\{(0, -1), (-3, 1)\}$	\mathcal{O}
10	$\{Q, \overline{Q}\}$	$\{\infty^+, \infty^+\}$
11	$\{(-3, 1), (-3, 1)\}$	$\{(0, 1), (0, 1)\}$

Table 1. The first 11 multiples of D_1 and \tilde{D}_1 .

It is apparent that $\pm D_1, \pm 7D_1, \pm 11D_1$ are all of the form $\{P, P\}$, and it is sufficient to show that no other member of $\mathcal{J}_1(\mathbb{Q})$ is of this form. Let $E_1 = 9D_1$, which is in the kernel of reduction mod 3 since $9\tilde{D}_1 = \mathcal{O}$, with corresponding local parameter $(-9/14, 426/49)$. Applying equation (28) we find that the local parameter of $n_1 E_1$ is $(36n_1, 3n_1 + 9n_1^3) \bmod 3^3$. Any $D \in \mathcal{J}_1(\mathbb{Q})$ can be written as $D = S + n_1 E_1$, for some $S \in \mathcal{S} = \{\mathcal{O}, D_1, 2D_1, \dots, 8D_1\}$. Consider, for example, $S = 2D_1$. Using the group law to compute (27) mod 3^3 at $D_0 = S = 2D_1$, and then substituting $(36n_1, 3n_1 + 9n_1^3)$ for (s_1, s_2) gives (31) as

$$\begin{aligned} \theta^{(1)}_{2D_1}(n_1) &\equiv 25 + 15n_1 + 18n_1^2 + 18n_1^3 \pmod{3^3}, \\ \theta^{(2)}_{2D_1}(n_1) &\equiv 6 + 24n_1 + 9n_1^2 + 18n_1^2 \pmod{3^3}, \\ \theta^{(3)}_{2D_1}(n_1) &\equiv 18n_1 + 18n_1^2 \pmod{3^3}, \end{aligned} \quad (33)$$

and so $\theta(2)_{2D_1}(n_1)^2 - 4\theta(1)_{2D_1}(n_1)\theta(3)_{2D_1}(n_1) \equiv 9 + 18n_1^2 \pmod{3^3}$. Strassmann's Theorem tells us that there are at most two roots. In fact we know that $n_1 = \pm 1$ are solutions, since $2D_1 + E_1 = 11D_1 = \{(-3, 1), (-3, 1)\}$ and $2D_1 - E_1 = -7D_1 = \{(0, 1), (0, 1)\}$ are both of the form $\{P, P\}$. Therefore, $n_1 = \pm 1$ are the only $n_1 \in \mathbb{Z}$ such that $2D_1 + n_1E_1$ is of the form $\{P, P\}$. Similar arguments show that: the only $n_1 \in \mathbb{Z}$ such that $D_1 + n_1E_1$ is of the form $\{P, P\}$ is $n_1 = 0$; the only $n_1 \in \mathbb{Z}$ such that $7D_1 + n_1E_1$ is of the form $\{P, P\}$ are $n_1 = 0, -2$; the only $n_1 \in \mathbb{Z}$ such that $8D_1 + n_1E_1$ is of the form $\{P, P\}$ is $n_1 = -1$. For the remaining five $S \in \mathcal{S}$, Strassmann's Theorem shows that $S + n_1E_1$ is never of this form. Hence the upper bound on the order of $\mathcal{C}_1(\mathbb{Q})$ is six, and so $\infty^\pm, (0, \pm 1), (-3, \pm 1)$ must give all of $\mathcal{C}_1(\mathbb{Q})$. \square

Combining Lemma 1 and Example 9 gives us the result shown in [14]

Theorem 3. *There is no quadratic polynomial in $\mathbb{Q}[z]$ with a rational point of exact period 5.*

A similar argument, but using the prime $p = 43$, shows that $\mathcal{C}_2(\mathbb{Q}) = \{\infty^\pm, (1, \pm 1)\}$, where \mathcal{C}_2 is as in (3) and Example 4 (which showed that $\mathcal{J}_2(\mathbb{Q})$ has rank 1). In view of Lemma 2, this gives a new proof of the result originally shown in [29] by an elaborate set of resultant and congruence arguments.

Theorem 4. *The only integer solutions to $a^2 + b^2 = c^2$, $a^3 + b^3 + c^3 = d^3$ are $(3, 4, 5, 6), (4, 3, 5, 6), (1, 0, -1, 0), (0, 1, -1, 0)$ up to scalar multiples.*

Both of the above examples are special cases of the following theorem of Chabauty [8].

Theorem 5. *Let \mathcal{C} be a curve of genus g defined over a number field K , whose Jacobian has Mordell-Weil rank $\leq g - 1$. Then \mathcal{C} has only finitely many K -rational points.*

Apparent from the above examples is the similarity between the strategy for finding all $(x, y) \in \mathcal{E}(K)$ with $x \in \mathbb{Q}$, where \mathcal{E} is an elliptic curve, $[K : \mathbb{Q}] = 2$, and $\mathcal{E}(K)$ has rank 1 (sometimes called "Elliptic Curve Chabauty"), and that for finding $\mathcal{C}(\mathbb{Q})$, where \mathcal{C} is a curve of genus 2 and $\mathcal{J}(\mathbb{Q})$ has rank 1. In each case, $\mathcal{E}(K)$ or $\mathcal{J}(\mathbb{Q})$, the group law is locally described by a 2-parameter system over \mathbb{Q} , and an arithmetic condition, $x \in \mathbb{Q}$ or $X_1 = X_2$, gives a power series in one variable n_1 . In general the local methods for finding all $(x, y) \in \mathcal{E}(K)$ with $x \in \mathbb{Q}$, where \mathcal{E} is an elliptic curve, $[K : \mathbb{Q}] = g$, and $\mathcal{E}(K)$ has rank less than g , will be similar to those for finding $\mathcal{C}(\mathbb{Q})$, where \mathcal{C} is a curve of genus g and $\mathcal{J}(\mathbb{Q})$ has rank less than g . Sometimes one can even choose between either of these to solve the same problem. The work done in Example 1 turns out to be equivalent to showing $\mathcal{F}_1(\mathbb{Q}) = \{\infty, (0, \pm 1)\}$ and $\mathcal{F}_2(\mathbb{Q}) = \{\infty\}$, where

$$\begin{aligned}\mathcal{F}_1 : t^2 &= (s^4 - 2s^2 - 8s + 1)(s^3 + s + 1), \\ \mathcal{F}_2 : \underline{t}^2 &= (\underline{s}^4 - 8\underline{s} - 4)(\underline{s}^3 + \underline{s}^2 + 1),\end{aligned}\tag{34}$$

both of genus 3. The derivation of $\mathcal{F}_1, \mathcal{F}_2$ will be made clear in the next section.

We conclude this section with the result in [2], which also makes use of Chabauty's Theorem.

Theorem 6. *The only $x, y, z \in \mathbb{Z}$ with $(x, y, z) = 1$, satisfying $x^2 + y^8 = z^3$ are $(\pm 1, 0, 1), (0, \pm 1, 1)$ and $(\pm 1549034, \pm 33, 15613)$.*

The proof uses a parametrisation of $x^2 + v^4 = z^3$ to obtain a covering of the solutions by the \mathbb{Q} -rational points on five curves of genus 2. Two of these can be resolved by maps to elliptic curves. The remaining three all have $\mathcal{J}(\mathbb{Q})$ of rank 1, and an argument similar to that used in the above examples can be used to find the rational points on each of them.

We should also mention that it is also possible to use differentials instead of the formal group as way of applying Chabauty's Theorem. This approach is described, for example, in [28]. For other work on Chabauty's Theorem, see also [9], [12], [18].

4 Coverings of Bielliptic Curves

We shall suppose, in this section, that our curve of genus 2 is defined over \mathbb{Q} and has a \mathbb{Q} -rational point, which has been mapped to infinity. Suppose also that there are only quadratic terms in X .

$$\mathcal{C} : Y^2 = G(X^2), \text{ where } G(x) = (x - e_1)(x - e_2)(x - e_3). \quad (35)$$

The map $X \mapsto -X$ swaps roots of the sextic of (35) in pairs, and the function $x = X^2$ is invariant under this map. There are then maps $(X, Y) \mapsto (X^2, Y)$ and $(X, Y) \mapsto (1/X^2, Y/X^3)$ from \mathcal{C} to the elliptic curves

$$\begin{aligned} \mathcal{E}^a : Y^2 &= G(x) = (x - e_1)(x - e_2)(x - e_3), \\ \mathcal{E}^b : \underline{Y}^2 &= \underline{x}^3 G(1/\underline{x}) = (-e_1 \underline{x} + 1)(-e_1 \underline{x} + 1)(-e_3 \underline{x} + 1), \end{aligned} \quad (36)$$

respectively. As in [28], these induce isogenies $\phi_1 : A_1 \rightarrow J$ and $\phi'_1 : J \rightarrow A_1$, where $A_1 = \mathcal{E}^a \times \mathcal{E}^b$.

$$\begin{aligned} \phi_1 : [(x, Y), (\underline{x}, \underline{Y})] &\mapsto \{(\sqrt{x}, Y), (-\sqrt{x}, Y)\} + \{(\frac{1}{\sqrt{\underline{x}}}, \frac{Y}{\underline{x}\sqrt{\underline{x}}}), (-\frac{1}{\sqrt{\underline{x}}}, -\frac{Y}{\underline{x}\sqrt{\underline{x}}})\}, \\ \phi'_1 : \{(X_1, Y_1), (X_2, Y_2)\} &\mapsto [(X_1^2, Y_1) + (X_2^2, Y_2), (\frac{1}{X_1^2}, \frac{Y_1}{X_1^3}) + (\frac{1}{X_2^2}, \frac{Y_2}{X_2^3})]. \end{aligned} \quad (37)$$

Both of ϕ_1, ϕ'_1 have kernels of order 4, and $\phi'_1 \circ \phi_1, \phi_1 \circ \phi'_1$ both give multiplication by 2 maps. There is furthermore an injective homomorphism (a special case of [20]):

$$\begin{aligned} \mu_1 : J(\mathbb{Q})/\phi_1(A_1(\mathbb{Q})) &\longrightarrow L_1^*/(L_1^*)^2 \times L_2^*/(L_2^*)^2 \times L_3^*/(L_3^*)^2 \\ : D &\mapsto [\mu_1^{(1)}(D), \mu_1^{(2)}(D), \mu_1^{(3)}(D)], \\ \text{where } \mu_1^{(j)} : \{(X_1, Y_1), (X_2, Y_2)\} &\mapsto (X_1^2 - e_j)(X_2^2 - e_j), \text{ for } j = 1, 2, 3, \end{aligned} \quad (38)$$

and where $L_i = \mathbb{Q}(e_i)$ for $i = 1, 2, 3$. This map is analogous to the map $(x, y) \mapsto x$ used to perform descent via 2-isogeny on an elliptic curve $y^2 = x(x^2 + ax + b)$ (see p.302 of [24]).

Suppose that, after performing a descent, we have determined the set

$$J(\mathbb{Q})/\phi_1(A_1(\mathbb{Q})) = \{D_1, \dots, D_m\}. \quad (39)$$

Let $(X, Y) \in \mathcal{C}(\mathbb{Q})$. Then $\{(X, Y), \infty^+\} = D_i$ in $J(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, for some $1 \leq i \leq m$, and so $\mu_1^{(j)}(\{(X, Y), \infty^+\}) = \mu^{(j)}(D_i)$ for $j = 1, 2, 3$, which is the same as $(X^2 - e_j) = \mu^{(j)}(D_i)$ in $L_j^*/(L_j^*)^2$ for $j = 1, 2, 3$. Since also $G(X^2)$ is a square by (35), we have

$$Y_{i,j}^2 = \mu^{(j)}(D_i)G(X^2)/(X^2 - e_j), \quad (40)$$

which is a curve of genus 1 defined over L_j (note that the right hand side is a quartic polynomial in X , after cancelling $X^2 - e_j$). Multiplying both sides by X^2 , we see that the variables $y_{i,j} = XY_{i,j}$ and $x = X^2$ satisfy

$$y_{i,j}^2 = \mu^{(j)}xG(x)/(x - e_j), \quad (41)$$

an elliptic curve isogenous to the Jacobian of (40). We now have a strategy for trying to find the \mathbb{Q} -rational points on the curve \mathcal{C} in (35), even when $\mathcal{J}(\mathbb{Q})$ has rank at least 2. Namely, for each i , one tries to find all $(x, y_{i,j})$ on (41) using the techniques at the beginning of Section 3. The following was proved first in [28] and then [15]. The proof we sketch here is a blend of those two proofs.

Theorem 7. *Let $\mathcal{C}_3 : Y^2 = X^6 + X^2 + 1$, the Diophantus curve of (5) and Example 5. Then $\mathcal{C}_3(\mathbb{Q}) = \{\infty^\pm, (0, \pm 1), (\pm 1/2, \pm 9/8)\}$.*

Proof We take $e_1 = \alpha$ where $\alpha^3 + \alpha + 1 = 0$, and note that $G(x) = x^3 + x + 1 = (x - \alpha)(x^2 + \alpha x + (\alpha^2 + 1))$. From Example 5 we know that $\mathcal{J}_3(\mathbb{Q})$ has rank 2 and is generated by $\{(0, 1), (0, 1)\}$ and $\{(0, 1), \infty^+\}$. We first note that $\{(0, 1), (0, 1)\}$ is trivial in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, as can be seen either by applying (37) to get $\{(0, 1), (0, 1)\} = \phi_1([(0, 1), \infty])$, or by applying (38) to get $\mu_1(\{(0, 1), (0, 1)\}) = [1, 1, 1]$. Applying (38) also gives that $\{(0, 1), \infty^+\} \neq \mathcal{O}$ in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$. We conclude that $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$ has exactly two members: $D_1 = \mathcal{O}$ and $D_2 = \{(0, 1), \infty^+\}$.

Let $(X, Y) \in \mathcal{C}_3(\mathbb{Q})$. Then $\{(X, Y), \infty^+\} = D_1$ or D_2 in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$. Applying (41) gives that $x = X^2 \in \mathbb{Q}$ satisfies one of the equations

$$\begin{aligned} y_{1,1}^2 &= x(x^2 + \alpha x + (\alpha^2 + 1)), \\ y_{2,1}^2 &= -\alpha x(x^2 + \alpha x + (\alpha^2 + 1)) \end{aligned} \quad (42)$$

We know from Example 7 that the only possible $x \in \mathbb{Q}$ are $x = \infty, 0, 1/4$, and so any $(X, Y) \in \mathcal{C}_3(\mathbb{Q})$ must satisfy $X = \infty, 0, \pm 1/2$, as required. \square

As an alternative, note that if $\{(X, Y), \infty^+\} = D_1 = \mathcal{O}$ in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$ then $\{(X, Y), \infty^+\} = \phi_1([R_a, R_b])$ for some $R_a \in \mathcal{E}^a(\mathbb{Q})$, $R_b \in \mathcal{E}^b(\mathbb{Q})$. Taking ϕ'_1 of both sides gives $[(X^2, Y) + \infty, (1/X^2, Y/X^3) + (0, 1)] = [2R_a, 2R_b]$. Let s be the x -coordinate of R_a , and let $[2]_a$ denote the x -coordinate duplication map on \mathcal{E}^a . Then

$$X^2 = [2]_a(s) = (s^4 - 2s^2 - 8s + 1)/4(s^3 + s + 1). \quad (43)$$

Letting $t = 2(s^3 + s + 1)X$ gives the model \mathcal{F}_1 in (34).

Similarly, if $\{(X, Y), \infty^+\} = D_2 = \{(0, 1), \infty^+\}$ in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$ then $\{(X, Y), \infty^+\} - D_2 = \{(X, Y), (0, -1)\} = \mathcal{O}$ in $J_3(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, so that

$\{(X, Y), (0, -1)\} = \phi_1([S_a, S_b])$ for some $S_a \in \mathcal{E}^a(\mathbb{Q})$, $S_b \in \mathcal{E}^b(\mathbb{Q})$. Taking ϕ'_1 of both sides gives $[(X^2, Y) + (0, -1), (1/X^2, Y/X^3) + \infty] = [2S_a, 2S_b]$. Let \underline{s} be the x -coordinate of S_b , and let $[2]_b$ denote the x -coordinate duplication map on \mathcal{E}^b . Then

$$1/X^2 = [2]_b(\underline{s}) = (\underline{s}^4 - 8\underline{s} - 4)/4(\underline{s}^3 + \underline{s}^2 + 1). \quad (44)$$

Letting $t = 2(\underline{s}^3 + \underline{s}^2 + 1)/X$ gives the model F_2 in (34). One can either, as we have done above, find all points the curves (42) with \mathbb{Q} -rational x -coordinate; or, as in [28], one can find all members of $F_1(\mathbb{Q}), F_2(\mathbb{Q})$.

The underlying geometry is described in [28]. Each D_i corresponds to an embedding of \mathcal{C} into its Jacobian, given by $P \mapsto \{P, \infty^+\} - D_i$. If the D_i give a complete set of representatives for $J(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, then every member of $\mathcal{C}(\mathbb{Q})$ will be ‘hit’ by $\phi_1(A(\mathbb{Q}))$ via one of these embeddings. It is therefore sufficient to find each $\mathcal{D}_i(\mathbb{Q})$, where \mathcal{D}_i is the pullback of the embedded curve. Each \mathcal{D}_i is a curve of genus 5 lying on A , and it has a hyperelliptic genus 3 quotient \mathcal{F}_i . In our example, these are the $\mathcal{F}_1, \mathcal{F}_2$ of (34). Furthermore, the Jacobians of $\mathcal{F}_1, \mathcal{F}_2$ are isogenous to the Weil restriction of scalars from $\mathbb{Q}(\alpha)$ to \mathbb{Q} of the curves in (42).

If we try solve $\mathcal{C}_4 : Y^2 = (X^2 + 15)(X^2 + 45)(X^2 + 135)$ of (6) by the same technique, a problem arises. Here, $e_1 = -15, e_2 = -45, e_3 = -135$, and every elliptic curve given by (41) is defined over \mathbb{Q} . This means that, if the method is to work, for every i at least one curve (41) for $j = 1, 2$ or 3 has to have rank 0. Applying (38) to the generators of $J_4(\mathbb{Q})$ given in Example 6, we find that the torsion group and $\{\infty^+, \infty^-\}$ are all trivial in $J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$. Hence $J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$ just consists of the two elements \mathcal{O} and $\{(3, 432), \infty^+\}$. Let $(X, Y) \in \mathcal{C}_4(\mathbb{Q})$. Then $\{(X, Y), \infty^+\}$ is equal to either $D_1 = \mathcal{O}$ or $D_2 = \{(3, 432), \infty^+\}$ in $J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$. Consider first the case $\{(X, Y), \infty^+\} = D_1 = \mathcal{O}$ in $J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$. Then, using (41), we know that $x = X^2$ satisfies $y_{1,1}^2 = x(x+45)(x+135)$, for some $y_{1,1} \in \mathbb{Q}$. This is an elliptic curve of rank 0 over \mathbb{Q} , which has only the 2-torsion points with $x = \infty, 0, -45, -135$. None of 0, -45, -135 are rational squares and so they do not correspond to points $(X, Y) \in \mathcal{C}(\mathbb{Q})$.

The case $\{(X, Y), \infty^+\} = D_2 = \{(3, 432), \infty^+\}$ is more troublesome. Using (41), we know that $x = X^2$ satisfies $y_{2,1}^2 = 24x(x+45)(x+135), y_{2,2}^2 = 54x(x+15)(x+135)$ and $y_{2,3}^2 = 144x(x+15)(x+45)$, for some $y_{2,1}, y_{2,2}, y_{2,3} \in \mathbb{Q}$. These are elliptic curves of ranks 2, 1, 1, respectively, over \mathbb{Q} , and so they do not restrict x to a finite number of choices. At this point, we have not determined $\mathcal{C}_4(\mathbb{Q})$, but we have shown

$$(X, Y) \in \mathcal{C}_4(\mathbb{Q}) \Rightarrow \{(X, Y), \infty^+\} = \{(3, 432), \infty^+\} \text{ in } J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q})). \quad (45)$$

For the curve \mathcal{C}_4 , the map $X \mapsto -X$ is not the only way of permuting the roots of the sextic. The curve is a special case of

$$Y^2 = (X^2 - k)(X^2 - rk)(X^2 - r^2k), \quad r, k \in \mathbb{Q}, \quad (46)$$

which has the involution $(X, Y) \mapsto (-rk/X, rk\sqrt{-rk} Y/X^3)$. The functions $U = (X + \sqrt{-rk})/(-X + \sqrt{-rk})$ and $V = (8\sqrt{-rk} Y)/(X - \sqrt{-rk})^3$ are invariant, and $(X, Y) \mapsto (U^2, V)$, $(X, Y) \mapsto (1/U^2, V/U^3)$ are maps from (46) to the quotient

$$v^2 = -2k(u+1)((r+1)^2 u^2 - 2(r^2 - 6r + 1)u + (r+1)^2), \quad (47)$$

defined over \mathbb{Q} . Viewing (47) as being defined over $\mathbb{Q}(\sqrt{-rk})$, let A_2 be its Weil-restriction over \mathbb{Q} . The maps $(X, Y) \mapsto (U^2, V)$, $(X, Y) \mapsto (1/U^2, V/U^3)$ induce isogenies $\phi_2 : A_2 \rightarrow J$ and $\phi'_2 : J \rightarrow A_2$, analogous to ϕ_1 of (37), where here J is the Jacobian of (46). There is also an injective homomorphism

$$\begin{aligned} \mu_2 : J(\mathbb{Q})/\phi_2(A_2(\mathbb{Q})) &\longrightarrow \mathbb{Q}^*/(\mathbb{Q}^*)^2 \times K^*/(K^*)^2, : D \mapsto [\mu_2^{(1)}(D), \mu_2^{(2)}(D)], \\ \mu_2^{(1)} : \{(X_1, Y_1), (X_2, Y_2)\} &\mapsto (X_1^2 - rk)(X_2^2 - rk), \\ \mu_2^{(2)} : \{(X_1, Y_1), (X_2, Y_2)\} &\mapsto (X_1 - \sqrt{k})(X_1 + r\sqrt{k})(X_2 - \sqrt{k})(X_2 + r\sqrt{k}), \end{aligned} \quad (48)$$

where $K = \mathbb{Q}(\sqrt{k})$. Suppose that, after performing a descent, we have determined the set

$$J(\mathbb{Q})/\phi_2(A_2(\mathbb{Q})) = \{D'_1, \dots, D'_n\}. \quad (49)$$

Let $(X, Y) \in \mathbb{Q}$. Then $\{(X, Y), \infty^+\} = D_i$ in $J(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, for some $1 \leq i \leq n$, and so $\mu_2^{(j)}(\{(X, Y), \infty^+\}) = \mu^{(j)}(D'_i)$ for $j = 1, 2$. By a similar argument to that used for (41), we can show (see [16] for details) that $u = 2X/(X^2 - rk)$ satisfies

$$y_i^2 = \mu_2^{(1)}(D'_i)\mu_2^{(2)}(D'_i)(rku^2 + 1)((r-1)\sqrt{k} u/2 + 1), \quad (50)$$

for some $y_i \in K$. If this is an elliptic curve of rank 1, then we can try to apply the Elliptic Curve Chabauty techniques described at the beginning of Section 3. For our curve C_4 of (6), a special case of (46) with $r = 3, k = -15$, we apply (38) to the generators of $J_4(\mathbb{Q})$ given in Example 6, and find that $\{\infty^+, \infty^+\}$ is trivial in $J_4(\mathbb{Q})/\phi_2(A_2(\mathbb{Q}))$. Hence $J_4(\mathbb{Q})/\phi_2(A_2(\mathbb{Q}))$ just consists of the eight elements generated by the 2-torsion and $\{(3, 432), \infty^+\}$, that is: $D'_1 = \mathcal{O}, D'_2 = \{(\beta, 0), (-\beta, 0)\}, D'_3 = \{(\sqrt{-45}, 0), (-\sqrt{-45}, 0)\}, D'_4 = \{(3\beta, 0), (-3\beta, 0)\}, D'_5 = \{(3, 432), \infty^+\}, D'_6 = D'_5 + D'_2, D'_7 = D'_5 + D'_3, D'_8 = D'_5 + D'_4$. Let $(X, Y) \in C_4(\mathbb{Q})$. Then $\{(X, Y), \infty^+\} = D'_i$ in $J_4(\mathbb{Q})/\phi_2(A_2(\mathbb{Q}))$ for some $1 \leq i \leq 8$. Now, for $i = 1, \dots, 4$, $D'_i = \mathcal{O}$ in $J_4(\mathbb{Q})/\phi_1(A_1(\mathbb{Q}))$, which has already been discounted by (45). For $i = 6, 7, 8$, one can use a straightforward 5-adic argument (see [16]) to show the nonexistence of $u \in \mathbb{Q}_5$, $y_i \in \mathbb{Q}_5(\beta)$, and hence the nonexistence of $u \in \mathbb{Q}, y_i \in \mathbb{Q}(\beta)$, satisfying (50).

In summary, if $(X, Y) \in C_4(\mathbb{Q})$, where C_4 is as in (6), then $\{(X, Y), \infty^+\} = D'_i$ in $J_4(\mathbb{Q})/\phi_2(A_2(\mathbb{Q}))$ for $i = 5$ or $i = 8$. Therefore $u = 2X/(X^2 - rk) = 2X/(X^2 + 45) \in \mathbb{Q}$ satisfies (50) for $i = 5$ or $i = 8$ (with $r = 3, k = -15$); that is, it satisfies one of the two equations

$$\begin{aligned} y_5^2 &= 6(54 + 6\beta)(-45u^2 + 1)(\beta u + 1), \\ y_8^2 &= 6(9 + \beta)(-45u^2 + 1)(\beta u + 1), \end{aligned} \quad (51)$$

for some y_5 or y_8 in $K = \mathbb{Q}(\beta) = \mathbb{Q}(\sqrt{-15})$. We have already seen, in Example 8, that the only $u \in \mathbb{Q}$ on either curve are $u = \infty, \pm 1/3, \pm 1/9$. For $u = \infty, \pm 1/3$,

there are no $X \in \mathbb{Q}$ satisfying $u = 2X/(X^2 + 45)$. For $u = \pm 1/9$, there are $X = \pm 3, \pm 15$; however, substituting $X = \pm 15$ into $(X^2 + 15)(X^2 + 45)(X^2 + 135)$ gives 23328000, which is nonsquare, and so there is no $(X, Y) \in \mathcal{C}(\mathbb{Q})$ with $X = \pm 15$. This leaves $X = \pm 3$ as the only possible X -coordinates of an affine $(X, Y) \in \mathcal{C}(\mathbb{Q})$. This proves that $\mathcal{C}(\mathbb{Q}) = \{\infty^\pm, (\pm 3, 432)\}$. In view of Lemma 3 this proves Conjecture 2, as in [16].

Theorem 8. *No polynomial of type $p_{3,1,1}$ is \mathbb{Q} -derived.*

A feature of the above proof is that covers via both ϕ_1 and ϕ_2 were required; neither the ϕ_1 nor the ϕ_2 information on its own is sufficient to determine $\mathcal{C}_4(\mathbb{Q})$.

5 Coverings of a General Curve of Genus 2

The next two sections use ideas of Nils Bruin, as in [2],[3], and variations by Flynn and Wetherell, as in [15],[17]. Let $\mathcal{C} : Y^2 = F(X) = F_1(X) \dots F_k(X)$ be a curve of genus 2, as in (1). We shall not assume that \mathcal{C} is of any of the special types in the last section, although we shall continue to assume that \mathcal{C} has a \mathbb{Q} -rational point that has been mapped to infinity. Let μ be the map on $\mathcal{J}(K)$ defined in (10), and suppose, as usual, that we have found $J(K)/2J(K)$. It is then straightforward to deduce $J(K)/\ker(\mu)$, which we list as

$$J(K)/\ker(\mu) = \{D_1, \dots, D_n\}. \quad (52)$$

Let $P = (X, Y) \in \mathcal{C}(K)$ so that $\{P, \infty^+\} \in J(K)$. Then, for some $i \in \{1, \dots, n\}$, we must have $\mu(\{(X, Y), \infty^+\}) = \mu(D_i)$. Let $G(x)$ be any polynomial of even degree such that $G(x)|F(x)$. Then there is an induced map

$$\mu_G : J(K) \rightarrow L_G^*/(L_G^*)^2 : [\sum_{j=1}^{\ell} n_j(x_j, y_j)] \mapsto \prod_{j=1}^{\ell} G(x_j)^{n_j}, \quad (53)$$

where L_G denotes the smallest field containing K over which $G(x)$ is defined. It follows that

$$q_G(D_i)G(x) \in (L_G^*)^2 \text{ for all } G(x)|F(x) \text{ with } 2|\deg(G(x)). \quad (54)$$

Each choice of G therefore gives a hyperelliptic curve $v_{i,G}^2 = q_G(D_i)G(x)$, defined over L_G , on which there must be an L_G -rational point with K -rational x -coordinate. When $G(x)$ has degree 4, it may be that this is an elliptic curve whose rank over L_G is less than $[L_G : K]$. In such cases, the Elliptic Curve Chabauty techniques at the beginning of Section 3 can be applied. This idea has recently been applied in [17] to $\mathcal{D} : x^4 + y^4 = 17$, the “Serre curve”, as in (7). This is a curve of genus 3 whose Jacobian has rank 6. It is shown on pp.187–189 of [7] that the rearrangement

$$(17 + (5x^2 - 4xy + 5y^2)) (17 - (5x^2 - 4xy + 5y^2)) = -2(2x^2 - 5xy + 2y^2)^2 \quad (55)$$

can be used, together with a resultant argument, to show that it is sufficient to find all \mathbb{Q} -rational points on the curve of genus 2

$$\mathcal{C}_5 : Y^2 = (9X^2 - 28X + 18)(X^2 + 12X + 2)(X^2 - 2). \quad (56)$$

Specifically, if $\mathcal{C}_5(\mathbb{Q})$ has no affine points, then $\mathcal{D}(\mathbb{Q})$ has only the affine points $(\pm 1, \pm 2), (\pm 2, \pm 1)$. Equations (7) and (56) have stubbornly resisted the techniques described in the last two sections, as well as the method of Dem'yanenko (see [21], p.67). However, [17] finally showed, using the ideas sketched above, that it is sufficient to find all points on an elliptic curve over $\mathbb{Q}(\sqrt{2}, \sqrt{17})$ with \mathbb{Q} -rational x -coordinate. This elliptic curve, which we do not reproduce here (see [17]) has rank 1 over $\mathbb{Q}(\sqrt{2}, \sqrt{17})$; the Elliptic Curve Chabauty techniques at the beginning of Section 3 can be applied to show that indeed $\mathcal{C}_5(\mathbb{Q})$ has no affine points, from which $\mathcal{D}(\mathbb{Q})$ can be deduced, as in [17].

Theorem 9. *The only $x, y \in \mathbb{Q}$ satisfying $x^4 + y^4 = 17$ are $(\pm 1, \pm 2), (\pm 2, \pm 1)$.*

The technique to obtain the genus 2 cover (56) generalises to other Fermat quartics $x^4 + y^4 = c$, and so the methods of [17] are potentially applicable to other nontrivial values of c ; that is, to the cases where $x^4 + y^4 = c$ cannot be trivially solved by a direct local argument or a map to a rank 0 elliptic curve. There are only four such cases with $c \leq 300$, namely: $c = 17, 82, 97, 257$.

6 A Classical Approach via Resultants

Given a curve such as

$$\mathcal{C}_6 : Y^2 = (X^2 + 1)(X^4 + 1), \quad (57)$$

one could, if desired, apply the techniques described above. Here, $\mathcal{J}_6(\mathbb{Q})$ has rank 2, and one can find $\mathcal{J}_6(\mathbb{Q})/2\mathcal{J}_6(\mathbb{Q})$, followed by a set of coverings curves as described in the last two sections. However, it is worth bearing in mind that more than enough techniques were available to deal with such a curve long before recent methods for finding $\mathcal{J}(\mathbb{Q})/2\mathcal{J}(\mathbb{Q})$. Letting $X = a/b$, where $a, b \in \mathbb{Z}$ and $\gcd(a, b) = 1$, and multiplying through by b^6 , we have that fg is an integer square, where $f = a^2 + b^2$ and $g = a^4 + b^4$. Now, if $d = \gcd(f, g)$ then d divides $g - (a^2 - b^2)f = 2b^4$ and $g + (a^2 - b^2)f = 2a^4$. Since $\gcd(a, b) = 1$, this means that $d|2$ and so $d = \pm 1, \pm 2$. Combining this with the fact that fg is an integer square gives that, for some choice of $d = \pm 1, \pm 2$, both of df and dg are integer squares. Dividing dg through by b^4 we have, in particular that $d(X^4 + 1)$ is a \mathbb{Q} -rational square, for some choice of $d = \pm 1, \pm 2$. The negative values of d give no such such $X \in \mathbb{R}$ and so no $X \in \mathbb{Q}$. This means that $(X, Y) \in \mathcal{C}_6(\mathbb{Q})$ satisfies $Y_1^2 = X^4 + 1$ for some $Y_1 \in \mathbb{Q}$ or $Y_2^2 = 2(X^4 + 1)$ for some $Y_2 \in \mathbb{Q}$. Both of these are rank 0 elliptic curves over \mathbb{Q} , the first having only the points $\infty^\pm, (0, \pm 1)$ and the second having only the points $(\pm 1, \pm 2)$, defined over \mathbb{Q} . We can therefore say that $\mathcal{C}_6(\mathbb{Q}) = \{\infty^\pm, (0, \pm 1), (\pm 1, \pm 2)\}$, without having done anything sophisticated.

In principle, this idea can be attempted even when $F(X)$ is written as a product of factors not defined over the ground field. When $F(X)$ is written as $F(X) = Q_1(X)Q_2(X)$, where $Q_1(X)$ is a quadratic and $Q_2(X)$ is a quartic, then resultant arguments (similar to those above) give a finite number of curves of genus 1 of the form: $y^2 = dQ_2(X)$, defined over an extension field, which need to be considered. One can then hope to apply Elliptic Curve Chabauty to each of these, and solve for $\mathcal{C}(\mathbb{Q})$ without ever having been required to compute $\mathcal{J}(\mathbb{Q})/2\mathcal{J}(\mathbb{Q})$. In [3], this strategy is used to solve the following Diophantine problem.

Theorem 10. *The only $x, y, z \in \mathbb{Z}$ with $(x, y, z) = 1$ and $xyz \neq 0$, satisfying $x^8 + y^3 = z^2$ are $(x, y, z) = (\pm 1, 2, \pm 3), (\pm 43, 96222, \pm 3004207)$.*

In the proof of this result, ten associated curves of genus 2 are found, as in Theorem 6. Of these, there are three difficult cases which required the technique outlined in this section, together with the Elliptic Curve Chabauty technique at the beginning of Section 3. It would also be possible to solve these three cases using the strategy in Section 5. It is, to some extent, a matter of taste. The resultant method in [3] bypasses the need to find $\mathcal{J}(\mathbb{Q})/2\mathcal{J}(\mathbb{Q})$. On the other hand, an initial computation of $\mathcal{J}(\mathbb{Q})/2\mathcal{J}(\mathbb{Q})$ is often a straightforward and efficient way of removing many of the curves $y^2 = dQ_2(X)$ from consideration.

The author thanks Nils Bruin, Bjorn Poonen and Michael Stoll for their helpful comments on an earlier draft of this manuscript.

References

1. C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. PARI-GP. Available from <ftp://megrez.math.u-bordeaux.fr/pub/pari>.
2. Nils Bruin. The Diophantine equations $x^2 \pm y^4 = \pm z^6$ and $x^2 + y^8 = z^3$. *Compositio Math.*, 118:305–321, 1999.
3. Nils Bruin. Chabauty methods using covers on curves of genus 2. Report MI 1999-15, Leiden. <http://www.math.leidenuniv.nl/reports/1999-15.shtml>
4. Nils Bruin. KASH-based program for performing 2-descent on elliptic curves over number fields. <http://www.math.uu.nl/people/bruin/ell.shar>
5. R.H. Buchholz and J.A. MacDougall. When Newton met Diophantus: A study of rational-derived polynomials and their extension to quadratic fields. To appear in *J. Number Theory*.
6. J.W.S. Cassels. *Local Fields*. LMS-ST 3. Cambridge University Press, Cambridge, 1986.
7. J.W.S. Cassels and E.V. Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2*. LMS-LNS 230. Cambridge University Press, Cambridge, 1996.
8. Claude Chabauty. Sur les points rationnels des variétés algébriques dont l’irrégularité est supérieure à la dimension. *C. R. Acad. Sci. Paris*, 212:1022–1024, 1941.
9. R.F. Coleman. Effective Chabauty, *Duke Math. J.*, 52:765–780, 1985.
10. M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, and K. Wildanger. KANT V4. *J. Symbolic Comput.*, 24(3-4):267–283, 1997. Available from <ftp://ftp.math.tu-berlin.de/pub/algebra/Kant/Kash>.
11. Z. Djabri, E.F. Schaefer, and N.P. Smart. Computing the p -Selmer group of an elliptic curve. Manuscript (1999). To appear in *Trans. Amer. Math. Soc.*

12. E.V. Flynn. A flexible method for applying chabauty's theorem. *Compositio Mathematica*, 105:79–94, 1997.
13. E.V. Flynn and N.P. Smart. Canonical heights on the Jacobians of curves of genus 2 and the infinite descent. *Acta Arith.*, 79:333–352, 1997.
14. E.V. Flynn, B. Poonen and E.F. Schaefer. Cycles of quadratic polynomials and rational points on a genus-two curve. *Duke Math. J.*, 90:435–463, 1997.
15. E.V. Flynn and J.L. Wetherell. Finding Rational Points on Bielliptic Genus 2 Curves. *Manuscripta Math.*, 100:519–533, 1999.
16. E.V. Flynn. On Q-Derived Polynomials. Manuscript (2000). To appear in *Proc. Edinburgh Math. Soc.*
17. E.V. Flynn and J.L. Wetherell. Covering Collections and a Challenge Problem of Serre. Manuscript (2000).
18. W. McCallum. On the method of Coleman and Chabauty. *Math. Ann.* 299(3): 565–596, 1994.
19. P. Morton. Arithmetic properties of periodic points of quadratic maps, II. *Acta Arith.* 87(2):89–102, 1998.
20. E.F. Schaefer. Computing a Selmer group of a Jacobian using functions on the curve. *Math. Ann.*, 310(3):447–471, 1998.
21. J.-P. Serre. *Lectures on the Mordell-Weil Theorem* Transl. and ed. by Martin Brown. From notes by Michel Waldschmidt. Wiesbaden; Braunschweig: Vieweg, 1989.
22. J. Sesiano. *Books IV to VII of Diophantus' Arithmetica in the Arabic Translation attributed to Qusta ibn Luqa*. Springer-Verlag, New York, 1982.
23. S. Siksek. Infinite descent on elliptic curves. *Rocky Mountain J. Math.*, 25(4):1501–1538, 1995.
24. J.H. Silverman. *The Arithmetic of Elliptic Curves*. GTM 106. Springer-Verlag, 1986.
25. M. Stoll. On the height constant for curves of genus two. *Acta Arith.* 90(2):183–201, 1999.
26. M. Stoll. Implementing 2-descent for Jacobians of hyperelliptic curves, preprint, 1999.
27. M. Stoll. On the height constant for curves of genus two, II. Manuscript (2000).
28. J.L. Wetherell. Bounding the Number of Rational Points on Certain Curves of High Rank. PhD Dissertation, University of California at Berkeley, 1997.
29. G.C. Young. On the Solution of a Pair of Simultaneous Diophantine Equations Connected with the Nuptial Numbers of Plato. *Proc. London Math. Soc.*, 23(2):27–44, 1924.

Lattice Reduction in Cryptology: An Update

Phong Q. Nguyen and Jacques Stern

École Normale Supérieure
Département d’Informatique
45 rue d’Ulm, 75005 Paris, France
[{Phong.Nguyen,Jacques.Stern}@ens.fr
<http://www.di.ens.fr/~{pnguyen,stern}/>](mailto:{Phong.Nguyen,Jacques.Stern}@ens.fr)

Abstract. Lattices are regular arrangements of points in space, whose study appeared in the 19th century in both number theory and crystallography. The goal of lattice reduction is to find useful representations of lattices. A major breakthrough in that field occurred twenty years ago, with the appearance of Lovász’s reduction algorithm, also known as LLL or L^3 . Lattice reduction algorithms have since proved invaluable in many areas of mathematics and computer science, especially in algorithmic number theory and cryptology. In this paper, we survey some applications of lattices to cryptology. We focus on recent developments of lattice reduction both in cryptography and cryptanalysis, which followed seminal works of Ajtai and Coppersmith.

1 Introduction

Lattices are discrete subgroups of \mathbb{R}^n . A lattice has infinitely many \mathbb{Z} -bases, but some are more useful than others. The goal of *lattice reduction* is to find interesting lattice bases, such as bases consisting of reasonably short and almost orthogonal vectors. From the mathematical point of view, the history of lattice reduction goes back to the reduction theory of quadratic forms developed by Lagrange [71], Gauss [44], Hermite [55], Korkine and Zolotarev [67,68], among others, and to Minkowski’s geometry of numbers [85]. With the advent of algorithmic number theory, the subject had a revival around 1980 with Lenstra’s celebrated work on integer programming (see [74]), which was, among others, based on a novel but non-polynomial time¹ lattice reduction technique. That algorithm inspired Lovász to develop a polynomial-time algorithm that computes a so-called *reduced* basis of a lattice. It reached a final form in the seminal paper [73] where Lenstra, Lenstra and Lovász applied it to factor rational polynomials in polynomial time (back then, a famous problem), from which the name LLL comes. Further refinements of the LLL algorithm were later proposed, notably by Schnorr [101,102].

Those algorithms have proved invaluable in many areas of mathematics and computer science (see [75,64,109,52,30,69]). In particular, their relevance to cryptology was immediately understood, and they were used to break schemes based

¹ The technique is however polynomial-time for fixed dimension, which was enough in [74].

on the knapsack problem (see [99,23]), which were early alternatives to the RSA cryptosystem [100]. The success of reduction algorithms at breaking various cryptographic schemes over the past twenty years (see [61]) have arguably established lattice reduction techniques as the most popular tool in public-key cryptanalysis. As a matter of fact, applications of lattices to cryptology have been mainly negative. Interestingly, it was noticed in many cryptanalytic experiments that LLL, as well as other lattice reduction algorithms, behave much more nicely than what was expected from the worst-case proved bounds. This led to a common belief among cryptographers, that lattice reduction is an easy problem, at least in practice.

That belief has recently been challenged by some exciting progress on the complexity of lattice problems, which originated in large part in two seminal papers written by Ajtai in 1996 and in 1997 respectively. Prior to 1996, little was known on the complexity of lattice problems. In his 1996 paper [3], Ajtai discovered a fascinating connection between the worst-case complexity and the average-case complexity of some well-known lattice problems. Such a connection is not known to hold for any other problem in NP believed to be outside P. In his 1997 paper [4], building on previous work by Adleman [2], Ajtai further proved the NP-hardness (under randomized reductions) of the most famous lattice problem, the shortest vector problem (SVP). The NP-hardness of SVP has been a long standing open problem. Ajtai's breakthroughs initiated a series of new results on the complexity of lattice problems, which are nicely surveyed by Cai [24,25].

Those complexity results opened the door to positive applications in cryptology. Indeed, several cryptographic schemes based on the hardness of lattice problems were proposed shortly after Ajtai's discoveries (see [5,49,56,26,83,41]). Some have been broken, while others seem to resist state-of-the-art attacks, for now. Those schemes attracted interest for at least two reasons: on the one hand, there are very few public-key cryptosystems based on problems different from integer factorization or the discrete logarithm problem, and on the other hand, some of those schemes offered encryption/decryption rates asymptotically higher than classical schemes. Besides, one of those schemes, by Ajtai and Dwork [5], enjoyed a surprising security proof based on worst-case (instead of average-case) hardness assumptions.

Independently of those developments, there has been renewed cryptographic interest in lattice reduction, following a beautiful work by Coppersmith [32] in 1996. Coppersmith showed, by means of lattice reduction, how to solve rigorously certain problems, apparently non-linear, related to the question of finding small roots of low-degree polynomial equations. In particular, this has led to surprising attacks on the celebrated RSA [100] cryptosystem in special settings such as low public or private exponent. Coppersmith's results differ from “traditional” applications of lattice reduction in cryptanalysis, where the underlying problem is already linear, and the attack often heuristic by requiring (at least) that current lattice reduction algorithms behave ideally, as opposed to what is theoretically guaranteed. The use of lattice reduction techniques to solve poly-

nomial equations goes back to the eighties [54,110]. The first result of that kind, the broadcast attack on low-exponent RSA due to Håstad [54], can be viewed as a weaker version of Coppersmith’s theorem on univariate modular polynomial equations.

The rest of the paper is organized as follows. In Section 2, we give basic definitions and results on lattices and their algorithmic problems. In Section 3, we survey an old topic of lattice reduction in cryptology, the well-known subset sum or knapsack problem. Subsequent sections cover more recent applications. In Section 4, we discuss lattice-based cryptography, somehow a revival for knapsack-based cryptography. In Section 5, we review the only positive application known of the LLL algorithm in cryptology, related to the hidden number problem. In Section 6, we discuss developments on the problem of finding small roots of polynomial equations, inspired by Coppersmith’s discoveries in 1996. In Section 7, we survey the surprising links between lattice reduction, the RSA cryptosystem, and integer factorization.

2 Lattice Problems

2.1 Definitions

Recall that a *lattice* is a discrete (additive) subgroup of \mathbb{R}^n . In particular, any subgroup of \mathbb{Z}^n is a lattice, and such lattices are called *integer lattices*. An equivalent definition is that a lattice consists of all integral linear combinations of a set of linearly independent vectors, that is,

$$L = \left\{ \sum_{i=1}^d n_i \mathbf{b}_i \mid n_i \in \mathbb{Z} \right\},$$

where the \mathbf{b}_i ’s are linearly independent over \mathbb{R} . Such a set of vectors \mathbf{b}_i ’s is called a lattice *basis*. All the bases have the same number $\dim(L)$ of elements, called the *dimension* (or *rank*) of the lattice.

There are infinitely many lattice bases. Any two bases are related to each other by some unimodular matrix (integral matrix of determinant ± 1), and therefore all the bases share the same Gram determinant $\det_{1 \leq i,j \leq d}(\mathbf{b}_i, \mathbf{b}_j)$. The *volume* $\text{vol}(L)$ (or *determinant*) of the lattice is by definition the square root of that Gram determinant, thus corresponding to the d -dimensional volume of the parallelepiped spanned by the \mathbf{b}_i ’s. In the important case of full-dimensional lattices where $\dim(L) = n$, the volume is equal to the absolute value of the determinant of any lattice basis (hence the name determinant). If the lattice is further an integer lattice, then the volume is also equal to the index $[\mathbb{Z}^n : L]$ of L in \mathbb{Z}^n .

Since a lattice is discrete, it has a shortest non-zero vector: the Euclidean norm of such a vector is called the lattice *first minimum*, denoted by $\lambda_1(L)$ or $\|L\|$. Of course, one can use other norms as well : we will use $\|L\|_\infty$ to denote the first minimum for the infinity norm. More generally, for all $1 \leq i \leq \dim(L)$, Minkowski’s i -th *minimum* $\lambda_i(L)$ is defined as the minimum of $\max_{1 \leq j \leq i} \|\mathbf{v}_j\|$

over all i linearly independent lattice vectors $\mathbf{v}_1, \dots, \mathbf{v}_i \in L$. It will be convenient to define the *lattice gap* as the ratio $\lambda_2(L)/\lambda_1(L)$ between the first two minima.

Minkowski's Convex Body Theorem guarantees the existence of short vectors in lattices: a careful application shows that any d -dimensional lattice L satisfies $\|L\|_\infty \leq \text{vol}(L)^{1/d}$, which is obviously the best possible bound. It follows that $\lambda_1(L) \leq \sqrt{d}\text{vol}(L)^{1/d}$, which is not optimal, but shows that the value $\lambda_1(L)/\text{vol}(L)^{1/d}$ is bounded when L runs over all d -dimensional lattices. The supremum of $\lambda_1(L)^2/\text{vol}(L)^{2/d}$ is denoted by γ_d , and called Hermite's constant² of dimension d , because Hermite was the first to establish its existence in the language of quadratic forms. The best asymptotic bounds known for Hermite's constant are the following ones (see [84, Chapter II] for the lower bound, and [31, Chapter 9] for the upper bound):

$$\frac{d}{2\pi e} + \frac{\log(\pi d)}{2\pi e} + o(1) \leq \gamma_d \leq \frac{1.744d}{2\pi e}(1 + o(1)).$$

Minkowski proved more generally:

Theorem 1 (Minkowski). *For all d -dimensional lattice L and all $r \leq d$:*

$$\prod_{i=1}^r \lambda_i(L) \leq \sqrt{\gamma_d^r} \text{vol}(L)^{r/d}.$$

More information on lattice theory can be found in numerous textbooks, such as [53, 108, 76].

2.2 Algorithmic Problems

In the rest of this section, we assume implicitly that lattices are rational lattices (lattices in \mathbb{Q}^n), and d will denote the lattice dimension.

The most famous lattice problem is the *shortest vector problem* (SVP), which was apparently first stated by Dirichlet in 1842: given a basis of a lattice L , find $\mathbf{v} \in L$ such that $\|\mathbf{v}\| = \lambda_1(L)$. SVP _{∞} will denote the analogue for the infinity norm. One defines approximate short vector problems by asking a non-zero $\mathbf{v} \in L$ with norm bounded by some approximation factor: $\|\mathbf{v}\| \leq f(d)\lambda_1(L)$.

The *closest vector problem* (CVP), also called the *nearest lattice point problem*, is a non-homogeneous version of the shortest vector problem: given a lattice basis and a vector $\mathbf{v} \in \mathbb{R}^n$, find a lattice vector minimizing the distance to \mathbf{v} . Again, one can define approximate versions.

Another problem is the *smallest basis problem* (SBP), which has many variants depending on the exact meaning of “smallest”. The variant currently in vogue (see [3, 11]) is the following: find a lattice basis minimizing the maximum of the lengths of its elements. A more geometric variant asks instead to minimize the product of the lengths (see [52]).

² For historical reasons, Hermite's constant refers to $\max \lambda_1(L)^2/\text{vol}(L)^{2/d}$ and not $\max \lambda_1(L)/\text{vol}(L)^{1/d}$.

2.3 Complexity Results

We refer to Cai [24,25] for an up-to-date survey of complexity results. Ajtai [4] recently proved that SVP is NP-hard under randomized reductions. Micciancio [82,81] simplified and improved the result by showing that approximating SVP to within a factor $< \sqrt{2}$ is also NP-hard under randomized reductions. The NP-hardness of SVP under deterministic (Karp) reductions remains an open problem.

CVP seems to be a more difficult problem. Goldreich *et al.* [50] recently noticed that CVP cannot be easier than SVP: given an oracle that approximates CVP to within a factor $f(d)$, one can approximate SVP in polynomial time to within the same factor $f(d)$. Reciprocally, Kannan proved in [64] that any algorithm approximating SVP to within a non-decreasing function $f(d)$ can be used to approximate CVP to within $d^{3/2}f(d)^2$. CVP was shown to be NP-hard as early as in 1981 [40] (for a simplified proof, see [65]). Approximating CVP to within a quasi-polynomial factor $2^{\log^{1-\varepsilon} d}$ is NP-hard [6,38].

However, NP-hardness results for SVP and CVP have limits. Goldreich and Goldwasser [46] showed that approximating SVP or CVP to within $\sqrt{d}/O(\log d)$ is not NP-hard, unless the polynomial-time hierarchy collapses.

Interestingly, SVP and CVP problems seem to be more difficult with the infinity norm. It was shown that SVP_∞ and CVP_∞ are NP-hard in 1981 [40]. In fact, approximating $\text{SVP}_\infty/\text{CVP}_\infty$ to within an almost-polynomial factor $d^{1/\log\log d}$ is NP-hard [37]. On the other hand, Goldreich and Goldwasser [46] showed that approximating $\text{SVP}_\infty/\text{CVP}_\infty$ to within $d/O(\log d)$ is not NP-hard, unless the polynomial-time hierarchy collapses.

We will not discuss Ajtai's worst-case/average-case equivalence [3,27], which refers to special versions of SVP and SBP (see [24,25,11]) such as SVP when the lattice gap λ_2/λ_1 is at least polynomial in the dimension.

2.4 Algorithmic Results

The main algorithmic results are surveyed in [75,64,109,52,30,69,24,97]. No polynomial-time algorithm is known for approximating either SVP, CVP or SBP to within a polynomial factor in the dimension d . In fact, the existence of such algorithms is an important open problem. The best polynomial time algorithms achieve only slightly subexponential factors, and are based on the LLL algorithm [73], which can approximate SVP and SBP. However, it should be emphasized that these algorithms typically perform much better than is theoretically guaranteed, on instances of practical interest. Given as input any basis of a lattice L , LLL provably outputs in polynomial time a basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ satisfying :

$$\|\mathbf{b}_1\| \leq 2^{(d-1)/4} \text{vol}(L)^{1/d}, \|\mathbf{b}_i\| \leq 2^{(d-1)/2} \lambda_i(L) \text{ and } \prod_{i=1}^d \|\mathbf{b}_i\| \leq 2^{\binom{d}{2}/2} \text{vol}(L).$$

Thus, LLL can approximate SVP to within $2^{(d-1)/2}$. Schnorr³ [101] improved the bound to $2^{O(d(\log \log d)^2 / \log d)}$. In fact, he defined an LLL-based family of algorithms [101] (named BKZ for blockwise Korkine-Zolotarev) whose performances depend on a parameter called the blocksize. These algorithms use some kind of exhaustive search exponential in the blocksize. So far, the best reduction algorithms in practice are variants [104,105] of those BKZ-algorithms, which apply a heuristic to reduce exhaustive search. But little is known on the average-case (and even worst-case) complexity of reduction algorithms.

Babai's nearest plane algorithm [7] uses LLL to approximate CVP to within $2^{d/2}$, in polynomial time (see also [66]). Using Schnorr's algorithm [101], this can be improved to $2^{O(d(\log \log d)^2 / \log d)}$, due to Kannan's link between CVP and SVP (see previous section). In practice however, the best strategy seems to be the *embedding method* (see [49,90]), which uses the previous algorithms for SVP and a simple heuristic reduction from CVP to SVP. Namely, given a lattice basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ and a vector $\mathbf{v} \in \mathbb{R}^n$, the embedding method builds the $(d+1)$ -dimensional lattice (in \mathbb{R}^{n+1}) spanned by the row vectors $(\mathbf{b}_i, 0)$ and $(\mathbf{v}, 1)$. It is hoped⁴ that a shortest vector of that lattice is of the form $(\mathbf{v} - \mathbf{u}, 1)$ where \mathbf{u} is a closest vector to \mathbf{v} , in the original lattice. Depending on the lattice, one should choose a coefficient different than 1 in $(\mathbf{v}, 1)$.

For exact SVP or CVP, the best algorithms known (in theory) are Kannan's super-exponential algorithms [63,65], with running time $2^{O(d \log d)}$.

3 Knapsacks

Cryptology and lattices share a long history with the *knapsack* (also called *subset sum*) problem, a well-known NP-hard problem considered by Karp: given a set $\{a_1, a_2, \dots, a_n\}$ of positive integers and a sum $s = \sum_{i=1}^n x_i a_i$, where $x_i \in \{0, 1\}$, recover the x_i 's.

In 1978, Merkle and Hellman[80] invented one of the first public-key cryptosystems, by converting some easy knapsacks into what they believed were hard knapsacks. It was basically the unique alternative to RSA until 1982, when Shamir [106] proposed an attack against the simplest version of the Merkle-Hellman scheme. Shamir used Lenstra's integer programming algorithm [74] but, the same year, Adleman [1] showed how to use LLL instead, making experiments much easier. Brickell [21,22] later extended the attacks to the more general “iterated” Merkle-Hellman scheme, and showed that Merkle-Hellman was insecure for all realistic parameters. The cryptanalysis of Merkle-Hellman schemes was the first application of lattice reduction in cryptology.

Despite the failure of Merkle-Hellman cryptosystems, researchers continued to search for knapsack cryptosystems because such systems are very easy to

³ Schnorr's result is usually cited in the literature as an approximation algorithm to within $(1+\varepsilon)^n$ for any constant $\varepsilon > 0$. However, Goldreich and Håstad noticed about a year ago that one can choose some $\varepsilon = o(1)$ and still have polynomial running time, for instance using the blocksize $k = \log d / \log \log d$ in [101].

⁴ Note that there exist simple counter-examples (see for instance [81]).

implement and can attain very high encryption/decryption rates. But basically, all knapsack cryptosystems have been broken (for a survey, see [99]), either by specific (often lattice-based) attacks or by the low-density attacks. The last significant candidate to survive was the Chor-Rivest cryptosystem [29], broken by Vaudenay [112] in 1997 with algebraic (not lattice) methods.

3.1 Low-Density Attacks

We only mention some of the links between lattices and knapsacks. Note that Ajtai's original proof [4] for the NP-hardness (under randomized reductions) of SVP used a connection between the subset sum problem and SVP.

The knapsack *density* is defined as $d = n / \max_{1 \leq i \leq n} \log_2 a_i$. The low-density attacks establish a reduction from the subset sum problem to the lattice shortest vector problem. The first low-density attack used the n -dimensional lattice $L(a_1, \dots, a_n, s)$ in \mathbb{Z}^{n+1} formed by the vectors (y_1, \dots, y_{n+1}) such that $y_1 a_1 + \dots + y_n a_n = y_{n+1} s$. Such a lattice can easily be built in polynomial time from the a_i 's and s . It was proved by Lagarias and Odlyzko [70] that if $d \leq 0.6463\dots$, the target vector $(x_1, \dots, x_n, 1)$ was the shortest vector of $L(a_1, \dots, a_n, s)$ with high probability over the choice of the a_i 's. The proof relies on bounds [77] on the number of integer points in n -dimensional balls. Thus, if one has access to an SVP-oracle, one can solve most subset sum problems of density $d \leq 0.6463\dots$. Coster *et al.* [34] later improved the connection between SVP and the knapsack problem. By using a simple variant of $L(a_1, \dots, a_n, s)$, they showed that if $d \leq 0.9408\dots$, the knapsack problem can be reduced to a lattice shortest vector problem (in dimension n) with high probability. In a different context (polynomial interpolation in the presence of noise), another example of attack based on provable reduction to SVP appeared recently in [10].

In the light of recent results on the complexity of SVP, those reductions from knapsack to SVP may seem useless. Indeed, the NP-hardness of SVP under randomized reductions suggests that there is no polynomial-time algorithm that solves SVP. However, it turns out that in practice, one can hope that standard lattice reduction algorithms behave like SVP-oracles, up to reasonably high dimensions. Experiments carried out in [70,104,105] show the effectiveness of such approach for solving low-density subset sums, up to n about the range of 100–200. It does not prove nor disprove that one can solve, in theory or in practice, low-density knapsacks with n over several hundreds. But it was sufficient to show that knapsack cryptography was impractical: indeed, the keysize of knapsack schemes grows in general at least quadratically with n , so that high values of n (as required by lattice attacks) are not practical.

One might wonder whether those reductions can lead to provable polynomial-time algorithms for certain subset sums. Recall that LLL is an SVP-oracle when the lattice gap is exponential in the lattice dimension. For lattices used in knapsack reductions, the gap increases as the knapsack density decreases, however the gap can be proved to be large enough only in extremely low density (see [42,43]). Hence, lattice methods to solve the subset sum problem are very heuristic. And lattice attacks against knapsack cryptosystems are somehow even more heuristic,

because the reductions from knapsack to SVP assume some (natural) property on the distribution of the weights a_i 's, which is in general not satisfied by knapsacks arising from cryptosystems.

3.2 The Orthogonal Lattice

Recently, Nguyen and Stern proposed in [91] a natural generalization of the Lagarias-Odlyzko [70] lattices. More precisely, they defined for any integer lattice L in \mathbb{Z}^n , the *orthogonal lattice* L^\perp as the set of integer vectors orthogonal to L , that is, the set of $\mathbf{x} \in \mathbb{Z}^n$ such that the dot product $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ for all $\mathbf{y} \in L$. Note that the lattice L^\perp has dimension $n - \dim(L)$, and can be computed in polynomial time from L (see [30]). Interestingly, the links between duality and orthogonality (see Martinet's book [76, pages 34–35]) enable to prove that the volume of L^\perp is equal to the volume of the intersection \bar{L} of \mathbb{Z}^n with the linear span of L . Thus, if a lattice in \mathbb{Z}^n is low-dimensional, its orthogonal lattice is high-dimensional with a volume at most equal: the successive minima of the orthogonal lattice are likely to be much shorter than the ones of the original lattice. That property of orthogonal lattices has led to effective (though heuristic) lattice-based attacks on various cryptographic schemes [91, 93, 94, 92, 95]. We refer to [96, 97] for more information. In particular, it was used in [95] to solve the *hidden subset sum problem* (used in [20]) in low density. The hidden subset sum problem was apparently a non-linear version of the subset sum problem: given M and n in \mathbb{N} , and $b_1, \dots, b_m \in \mathbb{Z}_M$, find $\alpha_1, \dots, \alpha_n \in \mathbb{Z}_M$ such that each b_i is some subset sum modulo M of $\alpha_1, \dots, \alpha_n$.

We sketch the solution of [95] to give a flavour of cryptanalyses based on orthogonal lattices. We first restate the hidden subset sum problem in terms of vectors. We are given an integer M , and a vector $\mathbf{b} = (b_1, \dots, b_m) \in \mathbb{Z}^m$ with entries in $[0..M-1]$ such that there exist integers $\alpha_1, \dots, \alpha_n \in [0..M-1]$, and vectors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{Z}^m$ with entries in $\{0, 1\}$ satisfying:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n \pmod{M}.$$

We want to determine the α_i 's. There exists a vector $\mathbf{k} \in \mathbb{Z}^m$ such that:

$$\mathbf{b} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n + M\mathbf{k}.$$

Notice that if \mathbf{u} in \mathbb{Z}^n is orthogonal to \mathbf{b} , then $\mathbf{p}_{\mathbf{u}} = (\langle \mathbf{u}, \mathbf{x}_1 \rangle, \dots, \langle \mathbf{u}, \mathbf{x}_n \rangle, \langle \mathbf{u}, \mathbf{k} \rangle)$ is orthogonal to the vector $\mathbf{v}_\alpha = (\alpha_1, \dots, \alpha_n, M)$. But \mathbf{v}_α is independent of m , and so is the n -dimensional lattice \mathbf{v}_α^\perp . On the other hand, as m grows for a fixed M , most of the vectors of any reduced basis of the $(m-1)$ -dimensional lattice \mathbf{b}^\perp should get shorter and shorter, because they should have norm close to $\text{vol}(\mathbf{b}^\perp)^{1/(m-1)} \leq \text{vol}(\mathbf{b})^{1/(m-1)} = \|\mathbf{b}\|^{1/(m-1)} \approx (M\sqrt{m})^{1/(m-1)}$. For such vectors \mathbf{u} , the corresponding vectors \mathbf{p}_u also get shorter and shorter. But if \mathbf{p}_u gets smaller than $\lambda_1(\mathbf{v}_\alpha^\perp)$ (which is independent of m), then it is actually zero, that is, \mathbf{u} is orthogonal to all the \mathbf{x}_j 's and \mathbf{k} . Note that one expects $\lambda_1(\mathbf{v}_\alpha^\perp)$ to be of the order of $\|\mathbf{v}_\alpha\|^{1/n} \approx (M\sqrt{n})^{1/n}$.

This suggests that if $(\mathbf{u}_1, \dots, \mathbf{u}_{m-1})$ is a sufficiently reduced basis of \mathbf{b}^\perp , then the first $m - (n + 1)$ vectors $\mathbf{u}_1, \dots, \mathbf{u}_{m-(n+1)}$ should heuristically be orthogonal to all the \mathbf{x}_j 's and \mathbf{k} . One cannot expect that more than $m - (n + 1)$ vectors are orthogonal because the lattice L_x spanned by the \mathbf{x}_j 's and \mathbf{k} is likely to have dimension $(n + 1)$. From the previous discussion, one can hope that the heuristic condition is satisfied when the density $n / \log(M)$ is very small (so that $\lambda_1(\mathbf{v}_\alpha^\perp)$ is not too small), and m is sufficiently large. And if the heuristic condition is satisfied, the lattice \bar{L}_x is disclosed, because it is then equal to the orthogonal lattice $(\mathbf{u}_1, \dots, \mathbf{u}_{m-(n+1)})^\perp$. Once \bar{L}_x is known, it is not difficult to recover (heuristically) the vectors \mathbf{x}_j 's by lattice reduction, because they are very short vectors. One eventually determines the coefficients α_j 's from a linear modular system. The method is quite heuristic, but it works in practice for small parameters in low density (see [95] for more details).

4 Lattice-Based Cryptography

We review state-of-the-art results on the main lattice-based cryptosystems. To keep the presentation simple, descriptions of the schemes are intuitive, referring to the original papers for more details. Only one of these schemes (the GGH cryptosystem [49]) explicitly works with lattices.

4.1 The Ajtai–Dwork Cryptosystem

Description. The Ajtai–Dwork cryptosystem [5] (AD) works in \mathbb{R}^n , with some finite precision depending on n . Its security is based on a variant of SVP.

The private key is a uniformly chosen vector u in the n -dimensional unit ball. One then defines a distribution \mathcal{H}_u of points \mathbf{a} in a large n -dimensional cube such that the dot product $\langle \mathbf{a}, \mathbf{u} \rangle$ is very close to \mathbb{Z} .

The public key is obtained by picking $\mathbf{w}_1, \dots, \mathbf{w}_n, \mathbf{v}_1, \dots, \mathbf{v}_m$ (where $m = n^3$) independently at random from the distribution \mathcal{H}_u , subject to the constraint that the parallelepiped w spanned by the \mathbf{w}_i 's is not flat. Thus, the public key consists of a polynomial number of points close to a collection of parallel affine hyperplanes, which is kept secret.

The scheme is mainly of theoretical purpose, as encryption is bit-by-bit. To encrypt a '0', one randomly selects b_1, \dots, b_m in $\{0, 1\}$, and reduces $\sum_{i=1}^m b_i \mathbf{v}_i$ modulo the parallelepiped w . The vector obtained is the ciphertext. The ciphertext of '1' is just a randomly chosen vector in the parallelepiped w . To decrypt a ciphertext \mathbf{x} with the private key u , one computes $\tau = \langle \mathbf{x}, u \rangle$. If τ is sufficiently close to \mathbb{Z} , then \mathbf{x} is decrypted as '0', and otherwise as '1'. Thus, an encryption of '0' will always be decrypted as '0', and an encryption of '1' has a small probability to be decrypted as '0'. These decryption errors can be removed (see [48]).

Security. The Ajtai–Dwork [5] cryptosystem received wide attention due to a surprising security proof based on worst-case assumptions. Indeed, it was shown

that any probabilistic algorithm distinguishing encryptions of a '0' from encryptions of a '1' with some polynomial advantage can be used to solve SVP in any n -dimensional lattice with gap λ_2/λ_1 larger than n^8 . There is a converse, due to Nguyen and Stern [93]: one can decrypt in polynomial time with high probability, provided an oracle that approximates SVP to within $n^{0.5-\varepsilon}$, or one that approximates CVP to within $n^{1.33}$. It follows that the problem of decrypting ciphertexts is unlikely to be NP-hard, due to the result of Goldreich-Goldwasser [46].

Nguyen and Stern [93] further presented a heuristic attack to recover the secret key. Experiments suggest that the attack is likely to succeed up to at least $n = 32$. For such parameters, the system is already impractical, as the public key requires 20 megabytes and the ciphertext for each bit has bit-length 6144. This shows that unless major improvements⁵ are found, the Ajtai-Dwork cryptosystem is only of theoretical importance.

Cryptanalysis Overview. At this point, the reader might wonder how lattices come into play, since the description of AD does not involve lattices. Any ciphertext of '0' is a sum of \mathbf{v}_i 's minus some integer linear combination of the \mathbf{w}_i 's. Since the parallelepiped spanned by the \mathbf{w}_i 's is not too flat, the coefficients of the linear combination are relatively small. On the other hand, any linear combination of the \mathbf{v}_i 's and the \mathbf{w}_i 's with small coefficients is close to the hidden hyperplanes. This enables to build a particular lattice of dimension $n + m$ such that any ciphertext of '0' is in some sense close to the lattice, and reciprocally, any point sufficiently close to the lattice gives rise to a ciphertext of '0'. Thus, one can decrypt ciphertexts provided an oracle that approximates CVP sufficiently well. The analogous version for SVP uses related ideas, but is technically more complicated. For more details, see [93].

The attack to recover the secret key can be described quite easily. One knows that each $\langle \mathbf{v}_i, \mathbf{u} \rangle$ is close to some unknown integer V_i . It can be shown that any sufficiently short linear combination of the \mathbf{v}_i 's give information on the V_i 's. More precisely, if $\sum_i \lambda_i \mathbf{v}_i$ is sufficiently short and the λ_i 's are sufficiently small, then $\sum_i \lambda_i V_i = 0$ (because it is a too small integer). Note that the V_i 's are disclosed if enough such equations are found. And each V_i gives an approximate linear equation satisfied by the coefficients of the secret key \mathbf{u} . Thus, one can compute a sufficiently good approximation of \mathbf{u} from the V_i 's. To find the V_i 's, we produce many short combinations $\sum_i \lambda_i \mathbf{v}_i$ with small λ_i 's, using lattice reduction. Heuristic arguments can justify that there exist enough such combinations. Experiments showed that the assumption was reasonable in practice.

4.2 The Goldreich–Goldwasser–Halevi Cryptosystem

The Goldreich-Goldwasser-Halevi cryptosystem [49] (GGH) can be viewed as a lattice-analog to the McEliece [78] cryptosystem based on algebraic coding theory. In both schemes, a ciphertext is the addition of a random noise vector

⁵ A variant of AD with less message expansion was proposed in [26], however without any security proof. It mixes AD with a knapsack.

to a vector corresponding to the plaintext. The public key and the private key are two representations of the same object (a lattice for GGH, a linear code for McEliece). The private key has a particular structure allowing to cancel noise vectors up to a certain bound. However, the domains in which all these operations take place are quite different.

Description. The GGH scheme works in \mathbb{Z}^n . The private key is a non-singular $n \times n$ integral matrix R , with very short row vectors⁶ (entries polynomial in n). The lattice L is the full-dimensional lattice in \mathbb{Z}^n spanned by the rows of R . The basis R is then transformed to a non-reduced basis B , which will be public. In the original scheme, B is the multiplication of R by sufficiently many small unimodular matrices. Computing a basis as “good” as the private basis R , given only the non-reduced basis B , means approximating SBP.

The message space is a “large enough” cube in \mathbb{Z}^n . A message $\mathbf{m} \in \mathbb{Z}^n$ is encrypted into $\mathbf{c} = \mathbf{m}B + \mathbf{e}$ where \mathbf{e} is an error vector uniformly chosen from $\{-\sigma, \sigma\}^n$, where σ is a security parameter. A ciphertext \mathbf{c} is decrypted as $\lfloor \mathbf{c}R^{-1} \rfloor RB^{-1}$ (note: this is Babai’s round method [7] to solve CVP). But an eavesdropper is left with the CVP-instance defined by \mathbf{c} and B . The private basis R is generated in such a way that the decryption process succeeds with high probability. The larger σ is, the harder the CVP-instances are expected to be. But σ must be small for the decryption process to succeed.

Improvements. In the original scheme, the public matrix B is the multiplication of the secret matrix by sufficiently many unimodular matrices. This means that without appropriate precaution, the public matrix can be as large as $O(n^3 \log n)$ bits.⁷ Micciancio [83] therefore suggested to define instead B as the Hermite normal form (HNF) of R . Recall that the HNF of an integer square matrix R in row notation is the unique lower triangular matrix with coefficients in \mathbb{N} such that: the rows span the same lattice as R , and any entry below the diagonal is strictly less than the diagonal entry in its column. Here, one can see that the HNF of R is $O(n^2 \log n)$ bits, which is much better but still big. When using the HNF, one should encode messages into the error vector \mathbf{e} instead of a lattice point, because the HNF is unbalanced. The ciphertext is defined as the reduction of \mathbf{e} modulo the HNF, and hence uses less than $O(n \log n)$ bits. One can easily prove that the new scheme (which is now deterministic) cannot be less secure than the original GGH scheme (see [83]).

Security. GGH has no proven worst-case/average-case property, but it is much more efficient than AD. Specifically, for security parameter n , key-size and encryption time can be $O(n^2 \log n)$ for GGH (McEliece is slightly better though),

⁶ A different construction for R based on tensor product was proposed in [41], but seems to worsen the decryption process.

⁷ Since the determinant has $O(n \log n)$ bits, one can always make the matrix smaller than $O(n^3 \log n)$ bits.

vs. at least $O(n^4)$ for AD. For RSA and El-Gamal systems, key size is $O(n)$ and computation time is $O(n^3)$. The authors of GGH argued that the increase in size of the keys was more than compensated by the decrease in computation time. To bring confidence in their scheme, they published on the Internet a series of five numerical challenges [47], in dimensions 200, 250, 300, 350 and 400. In each of these challenges, a public key and a ciphertext were given, and the challenge was to recover the plaintext.

The GGH scheme is now considered broken, at least in its original form, due to an attack recently developed by Nguyen [90]. As an application, using small computing power and Shoup’s NTL library [107], Nguyen was able to solve all the GGH challenges, except the last one in dimension 400. But already in dimension 400, GGH is not very practical: in the 400-challenge, the public key takes 1.8 Mbytes without HNF or 124 Kbytes using the HNF.⁸

Nguyen’s attack used two “qualitatively different” weaknesses of GGH. The first one is inherent to the GGH construction: the error vectors used in the encryption process are always much shorter⁹ than lattice vectors. This makes CVP-instances arising from GGH easier than general CVP-instances. The second weakness is the particular form of the error vectors in the encryption process. Recall that $\mathbf{c} = \mathbf{m}B + \mathbf{e}$ where $\mathbf{e} \in \{\pm\sigma\}^n$. The form of \mathbf{e} was apparently chosen to maximize the Euclidean norm under requirements on the infinity norm. However, by looking at that equation modulo some well-chosen integer (such as σ or even better, 2σ), it is possible to derive information on the message \mathbf{m} , which in turn leads to a simplification of the original closest vector problem, by shortening the error vector \mathbf{e} . The simplified closest vector problem happens to be within reach (in practice) of current lattice reduction algorithms, thanks to the embedding strategy that heuristically reduces CVP to SVP. We refer to [90] for more information.

It is easy to fix the second weakness by selecting the entries of the error vector \mathbf{e} at random in $[-\sigma \cdots + \sigma]$ instead of $\{\pm\sigma\}$. However, one can argue that the resulting GGH system would still be impractical, even using [83]. Indeed, Nguyen’s experiments [90] showed that SVP could be solved in practice up to dimensions as high as 350, for (certain) lattices with gap as small as 10. To be competitive, the new GGH system would require the hardness (in lower dimensions due to the size of the public key, even using [83]) of SVP for certain lattices of only slightly smaller gap, which means a rather smaller improvement in terms of reduction. Note also that those experiments do not support the practical hardness of Ajtai’s variant of SVP in which the gap is polynomial in the lattice dimension. Besides, it is not clear how to make decryption efficient without a huge secret key (Babai’s rounding requires the storage of R^{-1} or a good approximation, which could be in [49] over 1 Mbytes in dimension 400).

⁸ The challenges do not use the HNF, as they were proposed before [83]. Note that 124 Kbytes is about twice as large as McEliece for the recommended parameters.

⁹ In all GGH-like constructions known, the error vector is always at least twice as short. The situation is even worse in [41].

4.3 The NTRU Cryptosystem

Description. The NTRU cryptosystem [56], proposed by Hoffstein, Pipher and Silverman, works in the ring $R = \mathbb{Z}[X]/(X^N - 1)$. An element $F \in R$ is seen as a polynomial or a row vector: $F = \sum_{i=0}^{N-1} F_i x^i = [F_0, F_1, \dots, F_{N-1}]$. To select keys, one uses the set $\mathcal{L}(d_1, d_2)$ of polynomials $F \in R$ such that d_1 coefficients are equal to 1, d_2 coefficients are equal to -1, and the rest are zero. There are two small coprime moduli $p < q$: a possible choice is $q = 128$ and $p = 3$. There are also three integer parameters d_f, d_g and d_ϕ quite smaller than N (which is around a few hundreds).

The private keys are $f \in \mathcal{L}(d_f, d_f - 1)$ and $g \in \mathcal{L}(d_g, d_g)$. With high probability, f is invertible mod q . The public key $h \in R$ is defined as $h = g/f \pmod{q}$. A message $m \in \{-(p-1)/2 \dots + (p-1)/2\}^N$ is encrypted into: $e = (p\phi * h + m) \pmod{q}$, where ϕ is randomly chosen in $\mathcal{L}(d_\phi, d_\phi)$. The user can decrypt thanks to the congruence $e * f \equiv p\phi * g + m * f \pmod{q}$, where the reduction is centered (one takes the smallest residue in absolute value). Since ϕ, f, g and m all have small coefficients and many zeroes (except possibly m), that congruence is likely to be a polynomial equality over \mathbb{Z} . By further reducing $e * f$ modulo p , one thus recovers $m * f \pmod{q}$, hence m .

Security. The best attack known against NTRU is based on lattice reduction. The simplest lattice-based attack can be described as follows. Coppersmith and Shamir [33] noticed that the target vector $f \| g \in \mathbb{Z}^{2N}$ (the symbol $\|$ denotes vector concatenation) belongs to the following natural lattice:

$$L_{CS} = \{F \| G \in \mathbb{Z}^{2N} \mid F \equiv h * G \pmod{q} \text{ where } F, G \in R\}.$$

It is not difficult to see that L_{CS} is a full-dimensional lattice in \mathbb{Z}^{2N} , with volume q^N . The volume suggests that the target vector is a shortest vector of L_{CS} (but with small gap), so that a SVP-oracle should heuristically output the private keys f and g . However, based on numerous experiments with Shoup's NTL library [107], the authors of NTRU claimed in [56] that all such attacks are exponential in N , so that even reasonable choices of N ensure sufficient security. Note that the keysize of NTRU is only $O(N \log q)$, which makes NTRU the leading candidate among knapsack-based and lattice-based cryptosystems, and allows high lattice dimensions. It seems that better attacks or better lattice reduction algorithms are required in order to break NTRU. To date, none of the numerical challenges proposed in [56] has been solved. However, cryptographic concerns have been expressed about the lack of security proofs for NTRU: there is no known result proving that NTRU or variants of its encryption scheme satisfy standard security requirements (such as semantic security or non-malleability,¹⁰ see [79]), assuming the hardness of a sufficiently precise problem. Besides, there exist simple chosen ciphertext attacks [60] that can recover the secret key, so that appropriate padding is necessary.

¹⁰ NTRU without padding cannot be semantically secure since $e(1) \equiv m(1) \pmod{q}$ as polynomials. And it is easily malleable using multiplications by X of polynomials (circular shifts).

5 The Hidden Number Problem

5.1 Hardness of Diffie–Hellman Bits

There is only one example known in which the LLL algorithm plays a positive role in cryptology. In [18], Boneh and Venkatesan used LLL to solve the *hidden number problem*, which enables to prove the hardness of the most significant bits of secret keys in Diffie-Hellman and related schemes in prime fields. Recall the Diffie-Hellman key exchange protocol [36]: Alice and Bob fix a finite cyclic G and a generator g . They respectively pick random $a, b \in [1, |G|]$ and exchange g^a and g^b . The secret key is g^{ab} . Proving the security of the protocol under “reasonable” assumptions has been a challenging problem in cryptography (see [12]). Computing the most significant bits of g^{ab} is as hard as computing g^{ab} itself, in the case of prime fields:

Theorem 2 (Boneh–Venkatesan). *Let q be an n -bit prime and g be a generator of \mathbb{Z}_q^* . Let $\varepsilon > 0$ be fixed, and set $\ell = \ell(n) = \lceil \varepsilon \sqrt{n} \rceil$. Suppose there exists an expected polynomial time (in n) algorithm \mathcal{A} , that on input q, g, g^a and g^b , outputs the ℓ most significant bits of g^{ab} . Then there is also an expected polynomial time algorithm that on input q, g, g^a, g^b and the factorization of $q - 1$, computes all of g^{ab} .*

The above result is slightly different¹¹ from [18]. The same result holds for the least significant bits. For a more general statement when g is not necessarily a generator, and the factorization of $q - 1$ is unknown, see [51]. No such results are known for other groups (there is some kind of analogous result [113] for finite fields though).

The proof goes as follows. We are given some g^a and g^b , and want to compute g^{ab} . We repeatedly pick a random r until g^{a+r} is a generator of \mathbb{Z}_q^* (thanks to the factorization of $q - 1$). For each r , the probability of success is $\phi(q - 1)/(q - 1) \geq 1/\log \log q$. Next, we apply \mathcal{A} to the points g^{a+r} and g^{b+t} for many random values of t , so that we learn the most significant bits of $g^{(a+r)b} g^{(a+r)t}$, where $g^{(a+r)t}$ is a random element of \mathbb{Z}_q^* since g^{a+r} is a generator. Note that one can easily recover g^{ab} from $\alpha = g^{(a+r)b}$. The problem becomes the *hidden number problem* (HNP): given t_1, \dots, t_d chosen uniformly and independently at random in \mathbb{Z}_q^* , and $\text{MSB}_\ell(\alpha t_i \bmod q)$ for all i , recover $\alpha \in \mathbb{Z}_q$. Here, $\text{MSB}_\ell(x)$ for $x \in \mathbb{Z}_q$ denotes any integer z satisfying $|x - z| < q/2^\ell$.

To achieve the proof, Boneh and Venkatesan presented a simple solution to HNP when ℓ is not too small, by reducing HNP to a lattice closest vector problem. We sketch this solution in the next section. One can try to prove the hardness of Diffie-Hellman bits for different groups with the same method. Curiously, for the important case of elliptic curve groups, no efficient solution is known for the corresponding hidden number problem, except when one uses projective coordinates to represent elliptic curve points.

¹¹ Due to an error in the proof of [18] spotted by [51].

5.2 Solving the Hidden Number Problem by Lattice Reduction

Consider an HNP-instance: let t_1, \dots, t_d be chosen uniformly and independently at random in \mathbb{Z}_q^* , and $a_i = \text{MSB}_\ell(\alpha t_i \bmod q)$ where $\alpha \in \mathbb{Z}_q$ is hidden. Clearly, the vector $\mathbf{t} = (t_1 \alpha \bmod q, \dots, t_d \alpha \bmod q, \alpha/2^\ell)$ belongs to the $(d+1)$ -dimensional lattice $L = L(q, \ell, t_1, \dots, t_d)$ spanned by the rows of the following matrix:

$$\begin{pmatrix} q & 0 & \cdots & 0 & 0 \\ 0 & q & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & q & 0 \\ t_1 & \dots & \dots & t_d & 1/2^\ell \end{pmatrix}$$

The vector $\mathbf{a} = (a_1, \dots, a_d, 0)$ is very close to L , because it is very close to \mathbf{t} . Indeed, $\|\mathbf{t} - \mathbf{a}\| \leq q\sqrt{d+1}/2^\ell$. It is not difficult to show that any lattice point sufficiently close to \mathbf{a} discloses the hidden number α (see [18, Theorem 5] or [98]):

Lemma 3 (Uniqueness). *Set $d = 2\lceil\sqrt{\log q}\rceil$ and $\mu = \frac{1}{2}\sqrt{\log q} + 3$. Let α be in \mathbb{Z}_q^* . Choose integers t_1, \dots, t_d uniformly and independently at random in \mathbb{Z}_q^* . Let $\mathbf{a} = (a_1, \dots, a_d, 0)$ be such that $|(\alpha t_i \bmod q) - a_i| < q/2^\mu$. Then with probability at least $\frac{1}{2}$, all $\mathbf{u} \in L$ with $\|\mathbf{u} - \mathbf{a}\| < \frac{q}{2^\mu}$ are of the form:*

$$\mathbf{u} = (t_1 \beta \bmod q, \dots, t_d \beta \bmod q, \beta/2^\ell) \text{ where } \alpha \equiv \beta \pmod{q}.$$

Since \mathbf{a} is close enough to L , Babai's nearest plane CVP approximation algorithm [7] yields a lattice point sufficiently close to \mathbf{a} , which leads to:

Theorem 4 (Boneh-Venkatesan). *Let α be in \mathbb{Z}_q^* . Let \mathcal{O} be a function defined by $\mathcal{O}(t) = \text{MSB}_\ell(\alpha t \bmod q)$ with $\ell = \lceil\sqrt{\log q}\rceil + \lceil\log \log q\rceil$. There exists a deterministic polynomial time algorithm \mathcal{A} which, on input $t_1, \dots, t_d, \mathcal{O}(t_1), \dots, \mathcal{O}(t_d)$ outputs α with probability at least $1/2$ over t_1, \dots, t_d chosen uniformly and independently at random from \mathbb{Z}_q^* , where $d = 2\lceil\sqrt{\log q}\rceil$.*

Thus, the hidden number problem can be solved using $\ell = \sqrt{\log q} + \log \log q$ bits. Using Schnorr's improved lattice reduction algorithms, this can be asymptotically improved to $\varepsilon\sqrt{\log q}$ for any fixed $\varepsilon > 0$. One may also replace the bound $\frac{1}{2}$ by $\frac{1}{2\sqrt{n}}$ and reduce the number of bits required by $\log \log q$. Then, the expected run time goes up by a factor $\sqrt{\log q}$. One can alternately run $\sqrt{\log q}$ copies of the algorithm in parallel. Theorem 2 is a simple consequence.

5.3 Lattice Attacks on DSA

Interestingly, the previous solution of the hidden number problem also has a dark side: it leads to a simple attack against the Digital Signature Algorithm [88, 79] (DSA) in special settings (see [59, 98]). Recall that the DSA uses a public element $g \in \mathbb{Z}_p$ of order q , a 160-bit prime dividing $p-1$ where p is a large prime (at least

512 bits). The signer has a secret key $\alpha \in \mathbb{Z}_q^*$ and a public key $\beta = g^\alpha \bmod p$. The DSA signature of a message m is $(r, s) \in \mathbb{Z}_q^2$ where $r = (g^k \bmod p) \bmod q$, $s = k^{-1}(h(m) + \alpha r) \bmod q$, h is SHA-1 hash function and k is a random element in \mathbb{Z}_q^* chosen at each signature.

It is well-known that the secret key α can easily be recovered if the random nonce k is disclosed, or if k is produced by a cryptographically weak pseudo-random generator such as Knuth's linear congruential generator with known parameters [8]¹² and a few signatures are available. Recently, Howgrave-Graham and Smart [59] noticed that Babai's nearest plane algorithm could heuristically recover α , provided that sufficiently many signatures and sufficiently many bits of the corresponding nonces k are known. This is not surprising, because the underlying problem is in fact very close to the hidden number problem.

Indeed, assume that for d signatures (r_i, s_i) of messages m_i , the ℓ least significant bits of the random nonce k_i are known to the attacker: one knows $a_i < 2^\ell$ such that $k_i - a_i$ is of the form $2^\ell b_i$. Then $\alpha r_i \equiv s_i(a_i + b_i 2^\ell) - h(m_i) \pmod{q}$, which can be rewritten as: $\alpha r_i 2^{-\ell} s_i^{-1} \equiv (a_i - s_i^{-1} h(m_i)) \cdot 2^{-\ell} + b_i \pmod{q}$. Letting $t_i = r_i 2^{-\ell} s_i^{-1} \bmod q$, one sees that $\text{MSB}_\ell(at_i \bmod q)$ is known. Recovering the secret key α is therefore a slightly different hidden number problem in which the t_i 's are not assumed to be independent and uniformly distributed over \mathbb{Z}_q , but are of the form $r_i 2^{-\ell} s_i^{-1}$ where the underlying k_i 's are independent and uniformly distributed over \mathbb{Z}_q^* . In other words, HNP is an idealized version of the problem of breaking DSA (or related signature schemes) when the ℓ least significant bits (or more generally, ℓ consecutive bits) of the random nonce k are known for many signatures. It follows that Theorem 4 does not directly imply a provable attack on DSA in such settings.

But an attacker can ignore the difference between the distribution of $r_i 2^{-\ell} s_i^{-1}$ and the uniform distribution, and simply identify the DSA problem to HNP. Since lattice reduction algorithms can behave much better than theoretically expected, one can even hope to solve CVP exactly, yielding better bounds to Theorem 4. It is straightforward to extend Theorem 4 to the case where a CVP-oracle is available, by going through the proof of Lemma 3. For the case of a 160-bit prime q as in DSA, one obtains that HNP can be solved using respectively $\ell = 3$ bits and $d = 160$, or $\ell = 7$ bits and $d = 85$ respectively, when an oracle for CVP_∞ or CVP is available (see [98]). In fact, the bounds are even better in practice. It turns out that using standard lattice reduction algorithms implemented in Shoup's NTL library [107], one can often solve HNP for a 160-bit prime q using $\ell = 4$ bits and $d = 100$ (see [98]).

¹² Note that even in the simple case where the parameters of the linear congruential generator are hidden, the attack of [8] does not apply.

6 Finding Small Roots of Low-Degree Polynomial Equations

We survey an important application of lattice reduction found in 1996 by Coppersmith [32], and its developments. These results illustrate the power of linearization combined with lattice reduction.

6.1 Univariate Modular Equations

The general problem of solving univariate polynomial equations modulo some integer N of unknown factorization seems to be hard. Indeed, notice that for some polynomials, it is equivalent to the knowledge of the factorization of N . And the particular case of extracting e -th roots modulo N is the problem of decrypting ciphertexts in the RSA cryptosystem, for an eavesdropper. Curiously, Coppersmith [32] showed using LLL that the special problem of finding small roots is easy:

Theorem 5 (Coppersmith). *Let P be a monic polynomial of degree δ in one variable modulo an integer N of unknown factorization. Then one can find in time polynomial in $(\log N, 2^\delta)$ all integers x_0 such that $P(x_0) \equiv 0 \pmod{N}$ and $|x_0| \leq N^{1/\delta}$.*

Related (but weaker) results appeared in the eighties [54,110].¹³ We sketch a proof of Theorem 5, as presented by Howgrave-Graham [57], who simplified Coppersmith's original proof (see also [62]). Coppersmith's method reduces the problem of finding small modular roots to the (easy) problem of solving polynomial equations over \mathbb{Z} . More precisely, it applies lattice reduction to find an integral polynomial equation satisfied by all small modular roots of P . The intuition is to linearize all the equations of the form $x^i P(x)^j \equiv 0 \pmod{N^j}$ for appropriate integral values of i and j . Such equations are satisfied by any solution of $P(x) \equiv 0 \pmod{N}$. Small solutions x_0 give rise to unusually short solutions to the resulting linear system. To transform modular equations into integer equations, the following elementary lemma¹⁴ is used, with the notation $\|r(x)\| = \sqrt{\sum a_i^2}$ for any polynomial $r(x) = \sum a_i x^i \in \mathbb{Z}[x]$:

Lemma 6. *Let $r(x) \in \mathbb{Z}[x]$ be a polynomial of degree n and let X be a positive integer. Suppose $\|r(xX)\| < N^h / \sqrt{n}$. If $r(x_0) \equiv 0 \pmod{N^h}$ with $|x_0| < X$, then $r(x_0) = 0$ holds over the integers.*

Now the trick is to, given a parameter h , consider the $n = (h+1)\delta$ polynomials $q_{u,v}(x) = N^{h-v} x^u P(x)^v$, where $0 \leq u \leq \delta - 1$ and $0 \leq v \leq h$. Notice that any root x_0 of $P(x)$ modulo N is a root modulo N^h of $q_{u,v}(x)$, and therefore, of any integer linear combination $r(x)$ of the $q_{u,v}(x)$'s. If such a combination $r(x)$

¹³ Håstad [54] presented his result in terms of system of low-degree modular equations, but he actually studies the same problem, and his approach achieves the weaker bound $N^{2/(\delta(\delta+1))}$.

¹⁴ A similar lemma is used in [54]: the bound eventually obtained in [54] is weaker because only $h = 1$ is considered. Note also the resemblance with [73, Prop. 2.7].

further satisfies $\|r(xX)\| < N^h/\sqrt{n}$, then by Lemma 6, solving the equation $r(x) = 0$ over \mathbb{Z} yields all roots of $P(x)$ modulo N less than X in absolute value. This suggests to look for a short vector in the lattice corresponding to the $q_{u,v}(xX)$'s. More precisely, define the $n \times n$ matrix M whose i -th row consists of the coefficients of $q_{u,v}(xX)$, starting by the low-degree terms, where $v = \lfloor(i-1)/\delta\rfloor$ and $u = (i-1) - \delta v$. Notice that M is lower triangular, and a simple calculation leads to $\det(M) = X^{n(n-1)/2}N^{nh/2}$. We apply an LLL-reduction to the full-dimensional lattice spanned by the rows of M . The first vector of the reduced basis corresponds to a polynomial of the form $r(xX)$, and has Euclidean norm $\|r(xX)\|$. The theoretical bounds of the LLL algorithm ensure that:

$$\|r(xX)\| \leq 2^{(n-1)/4} \det(M)^{1/n} = 2^{(n-1)/4} X^{(n-1)/2} N^{h/2}.$$

Recall that we need $\|r(xX)\| \leq N^h/\sqrt{n}$ to apply the lemma. Hence, for a given h , the method is guaranteed to find modular roots up to X if:

$$X \leq \frac{1}{\sqrt{2}} N^{h/(n-1)} n^{-1/(n-1)}.$$

The limit of the upper bound, when h grows to ∞ , is $\frac{1}{\sqrt{2}} N^{1/\delta}$. Theorem 5 follows from an appropriate choice of h . This result is practical (see [35,58] for experimental results) and has many applications. It can be used to attack RSA encryption when a very low public exponent is used (see [13] for a survey). Boneh *et al.* [17] applied it to factor efficiently numbers of the form $N = p^r q$ for large r . Boneh [14] used a variant to find smooth numbers in short interval. See also [10] for an application to Chinese remainding in the presence of noise.

Remarks. Theorem 5 is trivial if P is monic. Note also that one cannot hope to improve the (natural) bound $N^{1/\delta}$ for all polynomials and all moduli N . Indeed, for the polynomial $P(x) = x^\delta$ and $N = p^\delta$ where p is prime, the roots of $P \pmod{N}$ are the multiples of p . Thus, one cannot hope to find all the small roots (slightly) beyond $N^{1/\delta} = p$, because there are too many of them. This suggests that even a SVP-oracle (instead of LLL) should not help Theorem 5 in general, as evidenced by the value of the lattice volume (the fudge factor $2^{(n-1)/4}$ yielded by LLL is negligible compared to $\det(M)^{1/n}$). It was recently noticed in [10] that if one only looks for the smallest root mod N , an SVP-oracle can improve the bound $N^{1/\delta}$ for very particular moduli (namely, squarefree N of known factorization, without too small factors). Note that in such cases, finding modular roots can still be difficult, because the number of modular roots can be exponential in the number of prime factors of N .

6.2 Multivariate Modular Equations

Interestingly, Theorem 5 can heuristically extend to multivariate polynomial modular equations. Assume for instance that one would like to find all small roots of $P(x, y) \equiv 0 \pmod{N}$, where $P(x, y)$ has total degree δ and has at

least one monic monomial $x^\alpha y^{\delta-\alpha}$ of maximal total degree. If one could obtain two algebraically independent integral polynomial equations satisfied by all sufficiently small modular roots (x, y) , then one could compute (by resultant) a univariate integral polynomial equation satisfied by x , and hence find efficiently all small (x, y) . To find such equations, one can use an analogue of lemma 6 to bivariate polynomials, with the (natural) notation $\|r(x, y)\| = \sum_{i,j} a_{i,j}^2$ for $r(x, y) = \sum_{i,j} a_{i,j} x^i y^j$:

Lemma 7. *Let $r(x, y) \in \mathbb{Z}[x, y]$ be a sum of at most w monomials. Assume $\|r(xX, yY)\| < N^h/\sqrt{w}$ for some $X, Y \geq 0$. If $r(x_0, y_0) \equiv 0 \pmod{N^h}$ with $|x_0| < X$ and $|y_0| < Y$, then $r(x_0, y_0) = 0$ holds over the integers.*

By analogy, one chooses a parameter h and select $r(x, y)$ as a linear combination of the polynomials $q_{u_1, u_2, v}(x, y) = N^{h-v} x^{u_1} y^{u_2} P(x, y)^v$, where $u_1 + u_2 + \delta v \leq h\delta$ and $u_1, u_2, v \geq 0$ with $u_1 < \alpha$ or $u_2 < \delta - \alpha$. Such polynomials have total degree less than $h\delta$, and therefore are linear combinations of the $n = (h\delta + 1)(h\delta + 2)/2$ monic monomials of total degree $\leq \delta h$. Due to the condition $u_1 < \alpha$ or $u_2 < \delta - \alpha$, such polynomials are in bijective correspondence with the n monic monomials (associate to $q_{u_1, u_2, v}(x, y)$ the monomial $x^{u_1+v\alpha} y^{u_2+v(\delta-\alpha)}$). One can represent the polynomials as n -dimensional vectors in such a way that the $n \times n$ matrix consisting of the $q_{u_1, u_2, v}(xX, yY)$'s (for some ordering) is lower triangular with coefficients $N^{h-v} X^{u_1+v\delta} y^{u_2+v(\delta-\alpha)}$ on the diagonal.

Now consider the first two vectors $r_1(xX, yY)$ and $r_2(xX, yY)$ of an LLL-reduced basis of the lattice spanned by the rows of that matrix. Since any root (x_0, y_0) of $P(x, y)$ modulo N is a root of $q_{u_1, u_2, v}(x, y)$ modulo N^h , we need $\|r_1(xX, yY)\|$ and $\|r_2(xX, yY)\|$ to be less than N^h/\sqrt{n} to apply Lemma 7. A (tedious) computation of the triangular matrix determinant enables to prove that $r_1(x, y)$ and $r_2(x, y)$ satisfy that bound when $XY < N^{1/\delta-\varepsilon}$ and h is sufficiently large (see [62]). Thus, one obtains two integer polynomial bivariate equations satisfied by all small modular roots of $P(x, y)$.

The problem is that, although such polynomial equations are linearly independent as vectors, they might be algebraically dependent, making the method heuristic. This heuristic assumption is unusual: many lattice-based attacks are heuristic in the sense that they require traditional lattice reduction algorithms to behave as SVP-oracles. An important open problem is to find sufficient conditions to make Coppersmith's method provable for bivariate (or multivariate) equations. Note that the method cannot work all the time. For instance, the polynomial $x - y$ has clearly too many roots over \mathbb{Z}^2 and hence too many roots mod any N (see [32] for more general counterexamples).

Such a result may enable to prove several attacks which are for now, only heuristic. Indeed, there are applications to the security of the RSA encryption scheme when a very low public exponent or a low private exponent is used (see [13] for a survey), and related schemes such as the KMOV cryptosystem (see [9]). In particular, the experimental evidence of [15,9] shows that the method is very effective in practice for certain polynomials.

Remarks. In the case of univariate polynomials, there was basically no choice over the polynomials $q_{u,v}(x) = N^{h-1-v}x^u P(x)^v$ used to generate the appropriate univariate integer polynomial equation satisfied by all small modular roots. There is much more freedom with bivariate modular equations. Indeed, in the description above, we selected the indices of the polynomials $q_{u_1,u_2,v}(x,y)$ in such a way that they corresponded to all the monomials of total degree $\leq h\delta$, which form a triangle in \mathbb{Z}^2 when a monomial $x^i y^j$ is represented by the point (i,j) . This corresponds to the general case where a polynomial may have several monomials of maximal total degree. However, depending on the shape of the polynomial $P(x,y)$ and the bounds X and Y , other regions of (u_1, u_2, v) might lead to better bounds.

Assume for instance $P(x,y)$ is of the form $x^{\delta_x} y^{\delta_y}$ plus a linear combination of $x^i y^j$'s where $i \leq \delta_x$, $j \leq \delta_y$ and $i + j < \delta_x + \delta_y$. Intuitively, it is better to select the (u_1, u_2, v) 's to cover the rectangle of sides $h\delta_x$ and $h\delta_y$ instead of the previous triangle, by picking all $q_{u_1,u_2,v}(x,y)$ such that $u_1 + v\delta_x \leq h\delta_x$ and $u_2 + v\delta_y \leq h\delta_y$, with $u_1 < \delta_x$ or $u_2 < \delta_y$. One can show that the polynomials $r_1(x,y)$ and $r_2(x,y)$ obtained from the first two vectors of an LLL-reduced basis of the appropriate lattice satisfy Lemma 7, provided that h is sufficiently large, and the bounds satisfy $X^{\delta_x} Y^{\delta_y} \leq N^{2/3-\varepsilon}$. Boneh and Durfee [15] applied similar and other tricks to a polynomial of the form $P(x,y) = xy + ax + b$. This allowed better bounds than the generic bound, leading to improved attacks on RSA with low secret exponent.

6.3 Multivariate Integer Equations

The general problem of solving multivariate polynomial equations over \mathbb{Z} is also hard, as integer factorization is a special case. Coppersmith [32] showed that a similar¹⁵ lattice-based approach can be used to find small roots of bivariate polynomial equations over \mathbb{Z} :

Theorem 8 (Coppersmith). *Let $P(x,y)$ be a polynomial in two variables over \mathbb{Z} , of maximum degree δ in each variable separately, and assume the coefficients of f are relatively prime as a set. Let X, Y be bounds on the desired solutions x_0, y_0 . Define $\hat{P}(x,y) = P(Xx, Yy)$ and let D be the absolute value of the largest coefficient of \hat{P} . If $XY < D^{2/(3\delta)}$, then in time polynomial in $(\log D, 2^\delta)$, we can find all integer pairs (x_0, y_0) such that $P(x_0, y_0) = 0$, $|x_0| < X$ and $|y_0| < Y$.*

Again, the method extends heuristically to more than two variables, and there can be improved bounds depending on the shape¹⁶ of the polynomial (see [32]). Theorem 8 was introduced to factor in polynomial time an RSA-modulus¹⁷ $N = pq$ provided that half of the (either least or most significant) bits of either

¹⁵ However current proofs are somehow more technical than for Theorem 5. A simplification analogue to what has been obtained for Theorem 5 would be useful.

¹⁶ The coefficient $2/3$ is natural from the remarks at the end of the previous section for the bivariate modular case. If we had assumed P to have total degree δ , the bound would be $XY < D^{1/\delta}$.

¹⁷ p and q are assumed to have similar size.

p or q are known (see [32,14,16]). This was sufficient to break an ID-based RSA encryption scheme proposed by Vanstone and Zuccherato [111]. Boneh *et al.* [16] provide another application, for recovering the RSA secret key when a large fraction of the bits of the secret exponent is known. Curiously, none of the applications cited above happen to be “true” applications of Theorem 8. It was later realized in [58,17] that those results could alternatively be obtained from a (simple) variant of the univariate modular case (Theorem 5).

7 Lattices and RSA

Section 6 suggests to clarify the links existing between lattice reduction and RSA [100], the most famous public-key cryptosystem. We refer to [79] for an exposition of RSA, and to [13] for a survey of attacks on RSA encryption. Recall that in RSA, one selects two prime numbers p and q of approximately the same size. The number $N = pq$ is public. One selects an integer d coprime with $\phi(N) = (p-1)(q-1)$. The integer d is the private key, and is called the RSA *secret exponent*. The *public exponent* is the inverse e of d modulo $\phi(N)$.

7.1 Lattice Attacks on RSA Encryption

Small Public Exponent. When the public exponent e is very small, such as 3, one can apply Coppersmith’s method (seen in the previous section) for univariate polynomials in various settings (see [13,32,35] for exact statements):

- An attacker can recover the plaintext of a given ciphertext, provided a large part of the plaintext is known.
- If a message is randomized before encryption, by simply padding random bits at a known place, an attacker can recover the message provided the amount of randomness is small.
- Håstad [54] attacks can be improved. An attacker can recover a message broadcasted (by RSA encryption and known affine transformation) to sufficiently many participants, each holding a different modulus N . This precisely happens if one sends a similar message with different known headers or time-stamps which are part of the encryption block.

None of the attacks recover the secret exponent d : they can only recover the plaintext. The attacks do not work if appropriate padding is used (see current standards and [79]), or if the public exponent is not too small. For instance, the popular choice $e = 65537$ is not threatened by these attacks.

Small Private Exponent. When $d \leq N^{0.25}$, an old result of Wiener [114] shows that one can easily recover the secret exponent d (and thus the factorization of N) from the continued fractions algorithm. Boneh and Durfee [15] recently improved the bound to $d \leq N^{0.292}$, by applying Coppersmith’s technique to bivariate modular polynomials and improving the generic bound. Note

that the attack is heuristic (see Section 6), but experiments showed that it works well in practice (no counterexample has ever been found). All those attacks on RSA with small private exponent also hold against the RSA signature scheme. A related result (using Coppersmith’s technique for either bivariate integer or univariate modular polynomials) is an attack [16] to recover d when a large portion of the bits of d is known (see [13]).

7.2 Lattice Attacks on RSA Signature

The RSA cryptosystem is often used as a digital signature scheme. To prevent various attacks, one must apply a preprocessing scheme to the message, prior to signature. The recommended solution is to use hash functions and appropriate padding (see current standards and [79]). However, several alternative simple solutions not involving hashing have been proposed, and sometimes accepted as standards. Today, all such solutions have been broken (see [45]), some of them by lattice reduction techniques (see [86,45]). Those lattice attacks are heuristic but work well in practice. They apply lattice reduction algorithms to find small solutions to (affine) linear systems, which leads to signature forgeries for certain proposed RSA signature schemes. Finding such small solutions is seen as a closest vector problem for some norm.

7.3 Factoring and Lattice Reduction

In the general case, the best attack against RSA encryption or signature is integer factorization. Note that to prove (or disprove) the equivalence between integer factorization and breaking RSA encryption remains an important open problem in cryptology (latest results [19] suggest that breaking RSA encryption may actually be easier). We already pointed out that in some special cases, lattice reduction leads to efficient factorization: when the factors are partially known [32], or when the number to factor has the form $p^r q$ with large r [17].

Schnorr [103] was the first to establish a link between integer factorization and lattice reduction, which was later extended by Adleman [2]. Schnorr [103] proposed a heuristic method to factor general numbers, using lattice reduction to approximate the closest vector problem in the infinity or the L_1 norm. Adleman [2] showed how to use the Euclidean norm instead, which is more suited to current lattice reduction algorithms. Those methods use the same underlying ideas as sieving algorithms (see [30]): to factor a number n , they try to find many congruences of smooth numbers to produce random square congruences of the form $x^2 \equiv y^2 \pmod{n}$, after a linear algebra step. Heuristic assumptions are needed to ensure the existence of appropriate congruences. The problem of finding such congruences is seen as a closest vector problem. Still, it should be noted that those methods are theoretical, since they are not adapted to currently known lattice reduction algorithms. To be useful, they would require very good lattice reduction for lattices of dimension over at least several thousands.

We close this review by mentioning that current versions of the Number Field Sieve (NFS) (see [72,30]), the best algorithm known for factoring large integers,

use lattice reduction. Indeed, LLL plays a crucial role in the last stage of NFS where one has to compute an algebraic square root of a huge algebraic number given as a product of hundreds of thousands of small ones. The best algorithm known to solve this problem is due to Montgomery (see [87,89]). It has been used in all recent large factorizations, notably the record factorization [28] of a 512-bit RSA-number of 155 decimal digits proposed in the RSA challenges. There, LLL is applied many times in low dimension (less than 10) to find nice algebraic integers in integral ideals. But the overall running time of NFS is dominated by other stages, such as sieving and linear algebra.

8 Conclusions

Lovász's algorithm and other lattice basis reduction algorithms have proved invaluable in cryptology. They have become the most popular tool in public-key cryptanalysis. In particular, they play a crucial role in several attacks against the RSA cryptosystem. The past few years have seen new, sometimes provable, lattice-based methods for solving problems which were *a priori* not linear, and this definitely opens new fields of applications. Paradoxically, at the same time, a series of complexity results on lattice reduction has emerged, giving rise to another family of cryptographic schemes based on the hardness of lattice problems. The resulting cryptosystems have enjoyed different fates, but it is probably too early to tell whether or not secure and practical cryptography can be built using hardness of lattice problems. Indeed, several questions on lattices remain open. In particular, we still do not know whether or not it is easy to approximate the shortest vector problem up to some polynomial factor, or to find the shortest vector when the lattice gap is larger than some polynomial in the dimension. Besides, only very few lattice basis reduction algorithms are known, and their behaviour (both complexity and output quality) is still not well understood. And so far, there has not been any massive computer experiment in lattice reduction comparable to what has been done for integer factorization or the elliptic curve discrete logarithm problem. Twenty years of lattice reduction yielded surprising applications in cryptology. We hope the next twenty years will prove as exciting.

Acknowledgements

We thank Dan Boneh, Glenn Durfee, Arjen Lenstra, László Lovász, Daniele Micciancio and Igor Shparlinski for helpful discussions and comments.

References

1. L. M. Adleman. On breaking generalized knapsack public key cryptosystems. In *Proc. of 15th STOC*, pages 402–412. ACM, 1983.
2. L. M. Adleman. Factoring and lattice reduction. Unpublished manuscript, 1995.
3. M. Ajtai. Generating hard instances of lattice problems. In *Proc. of 28th STOC*, pages 99–108. ACM, 1996. Available at [39] at TR96-007.

4. M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. of 30th STOC*. ACM, 1998. Available at [39] as TR97-047.
5. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. of 29th STOC*, pages 284–293. ACM, 1997. Available at [39] as TR96-065.
6. S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.
7. L. Babai. On Lovász lattice reduction and the nearest lattice point problem. *Combinatorica*, 6:1–13, 1986.
8. M. Bellare, S. Goldwasser, and D. Micciancio. "Pseudo-random" number generation within cryptographic algorithms: The DSS case. In *Proc. of Crypto '97*, volume 1294 of *LNCS*. IACR, Springer-Verlag, 1997.
9. D. Bleichenbacher. On the security of the KMOV public key cryptosystem. In *Proc. of Crypto '97*, volume 1294 of *LNCS*. IACR, Springer-Verlag, 1997.
10. D. Bleichenbacher and P. Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000.
11. J. Blömer and J.-P. Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proc. of 31st STOC*. ACM, 1999.
12. D. Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory – Proc. of ANTS-III*, volume 1423 of *LNCS*. Springer-Verlag, 1998.
13. D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, 1999.
14. D. Boneh. Finding smooth integers in short intervals using CRT decoding. In *Proc. of 32nd STOC*. ACM, 2000.
15. D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $n^{0.292}$. In *Proc. of Eurocrypt '99*, volume 1592 of *LNCS*, pages 1–11. IACR, Springer-Verlag, 1999.
16. D. Boneh, G. Durfee, and Y. Frankel. An attack on RSA given a small fraction of the private key bits. In *Proc. of Asiacrypt '98*, volume 1514 of *LNCS*, pages 25–34. Springer-Verlag, 1998.
17. D. Boneh, G. Durfee, and N. Howgrave-Graham. Factoring $N = p^r q$ for large r . In *Proc. of Crypto '99*, volume 1666 of *LNCS*. IACR, Springer-Verlag, 1999.
18. D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes. In *Proc. of Crypto '96*, LNCS. IACR, Springer-Verlag, 1996.
19. D. Boneh and R. Venkatesan. Breaking RSA may not be equivalent to factoring. In *Proc. of Eurocrypt '98*, volume 1233 of *LNCS*. IACR, Springer-Verlag, 1998.
20. V. Boyko, M. Peinado, and R. Venkatesan. Speeding up discrete log and factoring based schemes via precomputations. In *Proc. of Eurocrypt '98*, volume 1403 of *LNCS*, pages 221–235. IACR, Springer-Verlag, 1998.
21. E. F. Brickell. Solving low density knapsacks. In *Proc. of Crypto '83*. Plenum Press, 1984.
22. E. F. Brickell. Breaking iterated knapsacks. In *Proc. of Crypto '84*, volume 196 of *LNCS*. Springer-Verlag, 1985.
23. E. F. Brickell and A. M. Odlyzko. Cryptanalysis: A survey of recent results. In *Contemporary Cryptology*, pages 501–540. IEEE Press, 1991.
24. J.-Y. Cai. Some recent progress on the complexity of lattice problems. In *Proc. of FCRC*, 1999. Available at [39] as TR99-006.
25. J.-Y. Cai. The complexity of some lattice problems. In *Proc. of ANTS-IV*, LNCS. Springer-Verlag, 2000. In these proceedings.

26. J.-Y. Cai and T. W. Cusick. A lattice-based public-key cryptosystem. *Information and Computation*, 151:17–31, 1999.
27. J.-Y. Cai and A. P. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *Proc. of 38th FOCS*, pages 468–477. IEEE, 1997.
28. S. Cavallar, B. Dodson, A. K. Lenstra, W. Lioen, P. L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of 512-bit RSA key using the number field sieve. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000. Factorization announced in August, 1999.
29. B. Chor and R.L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Inform. Theory*, 34, 1988.
30. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1995. Second edition.
31. J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, 1998. Third edition.
32. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. of Cryptology*, 10(4):233–260, 1997. Revised version of two articles of Eurocrypt '96.
33. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Proc. of Eurocrypt '97*, LNCS. IACR, Springer-Verlag, 1997.
34. M.J. Coster, A. Joux, B.A. LaMacchia, A.M. Odlyzko, C.-P. Schnorr, and J. Stern. Improved low-density subset sum algorithms. *Comput. Complexity*, 2:111–128, 1992.
35. C. Coupé, P. Nguyen, and J. Stern. The effectiveness of lattice attacks against low-exponent RSA. In *Proc. of PKC'99*, volume 1431 of *LNCS*. Springer-Verlag, 1999.
36. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644–654, Nov 1976.
37. I. Dinur. Approximating SVP_{∞} to within almost-polynomial factors is NP-hard. Available at [39] at TR99-016.
38. I. Dinur, G. Kindler, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In *Proc. of 39th FOCS*, pages 99–109. IEEE, 1998. Available at [39] at TR98-048.
39. ECCC. <http://www.eccc.uni-trier.de/eccc/>. The Electronic Colloquium on Computational Complexity.
40. P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical report, Mathematisch Instituut, University of Amsterdam, 1981. Report 81-04. Available at <http://turing.wins.uva.nl/~peter/>.
41. R. Fischlin and J.-P. Seifert. Tensor-based trapdoors for CVP and their application to public key cryptography. In *Cryptography and Coding*, volume 1746 of *LNCS*, pages 244–257. Springer-Verlag, 1999.
42. A. M. Frieze. On the lagarias-odlyzko algorithm for the subset sum problem. *SIAM J. Comput*, 15(2):536–539, 1986.
43. M. L. Furst and R. Kannan. Succinct certificates for almost all subset sum problems. *SIAM J. Comput*, 18(3):550–558, 1989.
44. C.F. Gauss. *Disquisitiones Arithmeticae*. Leipzig, 1801.
45. M. Girault and J.-F. Misarsky. Cryptanalysis of countermeasures proposed for repairing ISO 9796–1. In *Proc. of Eurocrypt'2000*, LNCS. IACR, Springer-Verlag, 2000.
46. O. Goldreich and S. Goldwasser. On the limits of non-approximability of lattice problems. In *Proc. of 30th STOC*. ACM, 1998. Available at [39] as TR97-031.

47. O. Goldreich, S. Goldwasser, and S. Halevi. Challenges for the GGH cryptosystem. Available at <http://theory.lcs.mit.edu/~shaih/challenge.html>.
48. O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 105–111. IACR, Springer-Verlag, 1997. Available at [39] as TR97-018.
49. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 112–131. IACR, Springer-Verlag, 1997. Available at [39] as TR96-056.
50. O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. Available at [39] at TR99-002.
51. M. I. González Vasco and I. E. Shparlinski. On the security of Diffie-Hellman bits. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Comp. Number Theory (CCNT'99)*. Birkhauser, 2000.
52. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.
53. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers*. North-Holland, 1987.
54. J. Håstad. Solving simultaneous modular equations of low degree. *SIAM J. Comput.*, 17(2):336–341, April 1988. Early version in Proc. of Crypto '85.
55. C. Hermite. Extraits de lettres de M. Hermite à M. Jacobi sur différents objets de la théorie des nombres, deuxième lettre. *J. Reine Angew. Math.*, 40:279–290, 1850. Also in the first volume of Hermite's complete works (Gauthier-Villars).
56. J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a ring based public key cryptosystem. In *Proc. of ANTS III*, volume 1423 of *LNCS*, pages 267–288. Springer-Verlag, 1998. Additional information at <http://www.ntru.com>.
57. N. A. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and Coding*, volume 1355 of *LNCS*, pages 131–142. Springer-Verlag, 1997.
58. N. A. Howgrave-Graham. *Computational Mathematics Inspired by RSA*. PhD thesis, University of Bath, 1998.
59. N. A. Howgrave-Graham and N. P. Smart. Lattice attacks on digital signature schemes. Technical report, HP Labs, 1999. HPL-1999-90.
60. É. Jaulmes and A. Joux. A chosen ciphertext attack on NTRU. Preprint, 2000.
61. A. Joux and J. Stern. Lattice reduction: A toolbox for the cryptanalyst. *J. of Cryptology*, 11:161–185, 1998.
62. C. S. Jutla. On finding small solutions of modular multivariate polynomial equations. In *Proc. of Eurocrypt '98*, volume 1403 of *LNCS*, pages 158–170. IACR, Springer-Verlag, 1998.
63. R. Kannan. Improved algorithms for integer programming and related lattice problems. In *Proc. of 15th STOC*, pages 193–206. ACM, 1983.
64. R. Kannan. Algorithmic geometry of numbers. *Annual review of computer science*, 2:231–267, 1987.
65. R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.
66. P. Klein. Finding the closest lattice vector when it's unusually close. In *Proc. of SODA '2000*. ACM–SIAM, 2000.
67. A. Korkine and G. Zolotareff. Sur les formes quadratiques positives ternaires. *Math. Ann.*, 5:581–583, 1872.
68. A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Math. Ann.*, 6:336–389, 1873.
69. J. C. Lagarias. Point lattices. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 1, chapter 19. Elsevier, 1995.

70. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, January 1985.
71. L. Lagrange. Recherches d'arithmétique. *Nouv. Mém. Acad.*, 1773.
72. A. K. Lenstra and H. W. Lenstra, Jr. *The Development of the Number Field Sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, 1993.
73. A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Ann.*, 261:513–534, 1982.
74. H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.
75. L. Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50. SIAM, 1986. CBMS-NSF Regional Conference Series in Applied Mathematics.
76. J. Martinet. *Les Réseaux Parfaits des Espaces Euclidiens*. Editions Masson, 1996. English translation to appear at Springer-Verlag.
77. J. E. Mazo and A. M. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110:47–61, 1990.
78. R.J. McEliece. A public-key cryptosystem based on algebraic number theory. Technical report, Jet Propulsion Laboratory, 1978. DSN Progress Report 42-44.
79. A. Menezes, P. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
80. R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, IT-24:525–530, September 1978.
81. D. Micciancio. *On the Hardness of the Shortest Vector Problem*. PhD thesis, Massachusetts Institute of Technology, 1998.
82. D. Micciancio. The shortest vector problem is NP-hard to approximate within some constant. In *Proc. of 39th FOCS*. IEEE, 1998. Available at [39] at TR98-016.
83. D. Micciancio. Lattice based cryptography: A global improvement. Technical report, Theory of Cryptography Library, 1999. Report 99-05.
84. J. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Springer-Verlag, 1973.
85. H. Minkowski. *Geometrie der Zahlen*. Teubner-Verlag, Leipzig, 1896.
86. J.-F. Misarsky. A multiplicative attack using LLL algorithm on RSA signatures with redundancy. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 221–234. IACR, Springer-Verlag, 1997.
87. P. L. Montgomery. Square roots of products of algebraic numbers. In Walter Gautschi, editor, *Mathematics of Computation 1943–1993: a Half-Century of Computational Mathematics*, Proc. of Symposia in Applied Mathematics, pages 567–571. American Mathematical Society, 1994.
88. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 1994.
89. P. Nguyen. A Montgomery-like square root for the number field sieve. In *Proc. of ANTS-III*, volume 1423 of *LNCS*. Springer-Verlag, 1998.
90. P. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto '97. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 288–304. IACR, Springer-Verlag, 1999.
91. P. Nguyen and J. Stern. Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proc. of Crypto '97*, volume 1294 of *LNCS*, pages 198–212. IACR, Springer-Verlag, 1997.
92. P. Nguyen and J. Stern. Cryptanalysis of a fast public key cryptosystem presented at SAC '97. In *Selected Areas in Cryptography – Proc. of SAC '98*, volume 1556 of *LNCS*. Springer-Verlag, 1998.
93. P. Nguyen and J. Stern. Cryptanalysis of the Ajtai-Dwork cryptosystem. In *Proc. of Crypto '98*, volume 1462 of *LNCS*, pages 223–242. IACR, Springer-Verlag, 1998.
94. P. Nguyen and J. Stern. The Béguin-Quisquater server-aided RSA protocol from Crypto '95 is not secure. In *Proc. of Asiacrypt '98*, volume 1514 of *LNCS*, pages 372–379. Springer-Verlag, 1998.

95. P. Nguyen and J. Stern. The hardness of the hidden subset sum problem and its cryptographic implications. In *Proc. of Crypto '99*, volume 1666 of *LNCS*, pages 31–46. IACR, Springer-Verlag, 1999.
96. P. Nguyen and J. Stern. The orthogonal lattice: A new tool for the cryptanalyst. Manuscript submitted to *J. of Cryptology*, 2000.
97. P. Q. Nguyen. *La Géométrie des Nombres en Cryptologie*. PhD thesis, Université Paris 7, November 1999. Available at <http://www.di.ens.fr/~pnguyen/>.
98. P. Q. Nguyen. The dark side of the hidden number problem: Lattice attacks on DSA. In K.-Y. Lam, I. E. Shparlinski, H. Wang, and C. Xing, editors, *Proc. Workshop on Cryptography and Comp. Number Theory (CCNT'99)*. Birkhauser, 2000.
99. A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and Computational Number Theory*, volume 42 of *Proc. of Symposia in Applied Mathematics*, pages 75–88. A.M.S., 1990.
100. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. of the ACM*, 21(2):120–126, 1978.
101. C. P. Schnorr. A hierarchy of polynomial lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
102. C. P. Schnorr. A more efficient algorithm for lattice basis reduction. *J. of algorithms*, 9(1):47–62, 1988.
103. C. P. Schnorr. Factoring integers and computing discrete logarithms via diophantine approximation. In *Proc. of Eurocrypt '91*, volume 547 of *LNCS*, pages 171–181. IACR, Springer-Verlag, 1991.
104. C. P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66:181–199, 1994.
105. C. P. Schnorr and H. H. Hörmann. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Proc. of Eurocrypt '95*, volume 921 of *LNCS*, pages 1–12. IACR, Springer-Verlag, 1995.
106. A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In *Proc. of 23rd FOCS*, pages 145–152. IEEE, 1982.
107. V. Shoup. Number Theory C++ Library (NTL) version 3.9. Available at <http://www.shoup.net/ntl/>.
108. C. L. Siegel. *Lectures on the Geometry of Numbers*. Springer-Verlag, 1989.
109. B. Vallée. La réduction des réseaux: autour de l'algorithme de Lenstra, Lenstra, Lovász. *RAIRO Inform. Théor. Appl.*, 23(3):345–376, 1989. English translation in *CWI Quarterly*, 3(2): 95–120, 1990.
110. B. Vallée, M. Girault, and P. Toffin. How to guess ℓ -th roots modulo n by reducing lattice bases. In *Proc. of AAEEC-6*, volume 357 of *LNCS*, pages 427–442. Springer-Verlag, 1988.
111. S. A. Vanstone and R. J. Zuccherato. Short RSA keys and their generation. *J. of Cryptology*, 8(2):101–114, 1995.
112. S. Vaudenay. Cryptanalysis of the Chor-Rivest cryptosystem. In *Proc. of Crypto '98*, volume 1462 of *LNCS*. Springer-Verlag, 1998. Appeared first at the "rump session" of Crypto '97.
113. E. R. Verheul. Certificates of recoverability with scalable recovery agent security. In *Proc. of PKC '2000*, LNCS. Springer-Verlag, 2000.
114. M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Trans. Inform. Theory*, 36(3):553–558, 1990.

Construction of Secure C_{ab} Curves Using Modular Curves

Seigo Arita

C&C Media Research Laboratories, NEC
Kawasaki Kanagawa, Japan
arita@ccm.cl.nec.co.jp

Abstract. This paper proposes an algorithm which, given a basis of a subspace of the space of cuspforms of weight 2 for $\Gamma_0(N)$ which is invariant for the action of the Hecke operators, tests whether the subspace corresponds to a quotient A of the jacobian of the modular curve $X_0(N)$ such that A is the jacobian of a curve C . Moreover, equations for such a curve C are computed which make the quotient suitable for applications in cryptography. One advantage of using such quotients of modular jacobians is that fast methods are known for finding their number of points over finite fields [6]. Our results extend ideas of M. Shimura [13] who used only the full modular jacobian instead of abelian quotients of it.

1 C_{ab} Curve

First, we define C_{ab} curve following Miura[9]. Let C be an algebraic curve defined over a perfect field K with a place P of degree one. Take the ring $L(\infty P)$ of functions on C which are holomorphic away from P :

$$L(\infty P) = \{f \in K(C) \mid v_Q(f) \geq 0 \ (\forall Q \neq P)\}.$$

All of the pole numbers $-v_P(f)$ at P of $f \in L(\infty P)$ become a monoid M_P :

$$M_P = \{-v_P(f) \mid f \in L(\infty P)\}.$$

Take a minimum system $A = \{a_1, a_2, \dots, a_t\}$ ($a_1 < a_2 \dots < a_t$) of generators of M_P as a monoid:

$$M_P = \mathbb{N}_0 a_1 + \mathbb{N}_0 a_2 + \dots + \mathbb{N}_0 a_t = \langle A \rangle.$$

As M_P is co-finite in \mathbb{N}_0 , we have $\gcd(a_1, \dots, a_t) = 1$. For $A = \{a_1, \dots, a_t\}$, define a function Ψ_A on \mathbb{N}_0^t as

$$\Psi_A(n_1, \dots, n_t) = \sum_{i=1}^t a_i n_i \quad (n = (n_i) \in \mathbb{N}_0^t).$$

Definition 1 (C_{ab} Order). For $m = (m_1, \dots, m_t)$, and $n = (n_1, \dots, n_t) \in \mathbb{N}_0^t$, define an order $>_A$, as

$$m >_A n \stackrel{\text{def}}{\iff} \Psi_A(m) > \Psi_A(n)$$

or

$$\Psi_A(m) = \Psi_A(n), m_1 = n_1, \dots, m_{i-1} = n_{i-1}, m_i < n_i.$$

Then, the order $>_A$ becomes a monomial order, called “ C_{ab} order of type A ”. \square

We need to define two sets:

$$\begin{aligned} B(A) &= \{\text{the least } m \in \mathbb{N}_0^t \text{ w.r.t } C_{ab} \text{ order of type } A \text{ with } \Psi_A(m) = a \mid a \in \langle A \rangle\}, \\ V(A) &= \{l \in \mathbb{N}_0^t \setminus B(A) \mid l = m + n, m \in \mathbb{N}_0^t \setminus B(A), n \in \mathbb{N}_0^t \Rightarrow n = (0, 0, \dots, 0)\}. \end{aligned}$$

Miura[9] showed

Theorem 1. Let C be an algebraic curve defined over a perfect field K with a place P of degree one. Then, if

$$M_P = \langle A \rangle, \quad A = \{a_1, \dots, a_t\}, \quad a_1 < \dots < a_t$$

holds, the curve C has a nonsingular affine model in t dimensional affine space with the defining equations

$$F_m = X^m + \alpha_l X^l + \sum_{n \in B(A), \Psi_A(n) < \Psi_A(m)} \alpha_n X^n \quad (m \in V(A)). \quad (1)$$

Here, l is a unique $l \in B(A)$ satisfying $\Psi_A(m) = \Psi_A(l)$, and $\alpha_l \neq 0, \alpha_n \in K$.

The affine curve $F_m = 0$ ($m \in V(A)$), obtained from $A = \{a_1, \dots, a_t\}$ ($\gcd(a_1, \dots, a_t) = 1, a_1 < \dots < a_t$), is called a “ C_{ab} curve of type A ”.

Example: $C_{3,5,7}$ Curve

$C_{3,5,7}$ curve, that is C_{ab} curve of type $\{3, 5, 7\}$, is a space curve defined by three equations of the form:

$$\begin{aligned} Y^2 &= a_0 XZ + a_1 X^3 + a_2 XY + a_3 Z + a_4 X^2 + a_5 Y + a_6 X + a_7 \\ YZ &= b_0 X^4 + b_1 X^2 Y + b_2 XZ + b_3 X^3 + b_4 XY + b_5 Z + b_6 X^2 \\ &\quad + b_7 Y + b_8 X + b_9 \\ Z^2 &= c_0 X^3 Y + c_1 X^2 Z + c_2 X^4 + c_3 X^2 Y + c_4 XZ + c_5 X^3 + c_6 XY \\ &\quad + c_7 Z + c_8 X^2 + c_9 Y + c_{10} X + c_{11}. \end{aligned} \quad (2)$$

2 Security Condition

A discrete log based cryptosystem using the jacobian of a curve C over a field \mathbb{F}_q will be less secure than a standard 1024 bit RSA system, unless four conditions are satisfied.

1. (Against Pollard's rho algorithm)

The order h of the Jacobian J_C has a prime factor l of 160 or more bits[10].

2. (Against FR attack)

The prime factor l does not divide $q^k - 1$ for small k [5].

3. (Against Rück attack)

l should be coprime with q [12].

4. (Against Gaudry's variant)

q has $(40 + \log_2(84(g - 1)) + \log_2(m))$ or more bits[7,3].

3 Construction of Secure C_{ab} Curves

For definitions of the congruence subgroup $\Gamma_0(N)$, the modular curve $X_0(N)$, and so on, see [4] or [11].

3.1 Number of Points of a Simple Factor of a Modular Curve

Let N be a natural number. Let $C(N)$ be a \mathbb{Q} -vector space with basis $\mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$. Let $B(N)$ be the subspace of $C(N)$ spanned by all elements of the form

$$(c : d) + (-d : c), \\ (c : d) + (c + d : -c) + (d : -c - d).$$

Let $C_0(N)$ be a \mathbb{Q} -vector space spanned by $\Gamma_0(N)$ -cusps. Define the boundary map $\delta : C(N) \rightarrow C_0(N)$ by

$$\delta((c : d)) = [a/c] - [b/d]$$

where integers a and b are chosen so that $ad - bc = 1$, and set $Z(N) = \ker(\delta)$. Note that $B(N) \subset Z(N)$. Finally, define $H(N) = Z(N)/B(N)$, which is a \mathbb{Q} -vector space of dimension $2g$, where g is the genus of the modular curve $X_0(N)$.

Let $H^+(N)$ be a +1 proper space of the star operator $* : (c : d) \mapsto (-c : d)$ on $H(N)$.

Proposition 1 ([4]). *Let \mathbb{T} be the Hecke algebra of level N . As \mathbb{T} -modules,*

$$H^+(N) \otimes_{\mathbb{Q}} \mathbb{C} \simeq S_2(N).$$

□

Hecke operator T_p can be dealt with as an operator on $H^+(N)$. For a prime p not dividing N , the operator T_p on $H^+(N)$ is calculated by Heilbronn matrices R_p :

$$T_p((c : d)) = \sum_{M \in R_p} (c : d) M.$$

An algorithm for computing Heilbronn matrices is given in page 22 of [4].

A simple factor A of the Jacobian $J_0(N)$ corresponds to a simple \mathbb{T} -submodule K of $H^+(N)$ one-to-one. The dimension of A as an abelian variety is equal to the dimension of K as a vector space over \mathbb{Q} . Using Eichler-Shimura relation [11] one finds the formula for the number of points on a factor A over a prime field \mathbb{F}_p for a prime p not dividing N ,

$$\#A/\mathbb{F}_p = \text{Det}(x^2 - T_p|_K x + p)|_{x=1}. \quad (3)$$

Example: Level 97. Let $N = 97$. $H^+(97)$ is 7-dimensional over \mathbb{Q} with a basis $\{g_1, g_2, \dots, g_7\}$:

$$\begin{aligned} g_1 &= (44 : 1) - (88 : 1) + (91 : 1), \\ g_2 &= (70 : 1) + (87 : 1) - (88 : 1) + (91 : 1) - (92 : 1) + (93 : 1) - (94 : 1), \\ g_3 &= (78 : 1) - (92 : 1) + (93 : 1) - (94 : 1), \\ g_4 &= (79 : 1) + (87 : 1) - (88 : 1) + (91 : 1) - (92 : 1) + (93 : 1) - (94 : 1), \\ g_5 &= (83 : 1) - (90 : 1), \\ g_6 &= (89 : 1) - (91 : 1), \\ g_7 &= (95, 1). \end{aligned}$$

Calculating the characteristic polynomial of T_2 using the basis $\{g_1, g_2, \dots, g_7\}$ and factoring it over \mathbb{Q} , we get

$$(-1 + 3x + 4x^2 + x^3)(-1 + 6x - x^2 - 3x^3 + x^4). \quad (4)$$

This leads to the guess that $J_0(97)$ is factored to the product of 3-dimensional simple abelian variety A_3 and 4-dimensional simple abelian variety A_4 .

Let K_3 be the \mathbb{T} -submodule of $H^+(97)$ corresponding to A_3 . K_3 is spanned by proper vectors of the irreducible factor $-1 + 3x + 4x^2 + x^3$ of Equation (4). Using this, we can find a basis $\{f_1, f_2, f_3\}$ of K_3 over \mathbb{Q} :

$$\begin{aligned} f_1 &= 8 \cdot (44 : 1) + 30 \cdot (70 : 1) - 16 \cdot (78 : 1) + 2 \cdot (79 : 1) + 20 \cdot (83 : 1) \\ &\quad + 32 \cdot (87 : 1) - 40 \cdot (88 : 1) + 35 \cdot (89 : 1) - 20 \cdot (90 : 1) + 5 \cdot (91 : 1) \\ &\quad - 16 \cdot (92 : 1) + 16 \cdot (93 : 1) - 16 \cdot (94 : 1) + 39 \cdot (95 : 1), \\ f_2 &= -6 \cdot (44 : 1) - 68 \cdot (70 : 1) + 12 \cdot (78 : 1) + 44 \cdot (79 : 1) - 15 \cdot (83 : 1) \\ &\quad - 24 \cdot (87 : 1) + 30 \cdot (88 : 1) - 49 \cdot (89 : 1) + 15 \cdot (90 : 1) + 19 \cdot (91 : 1) \\ &\quad + 12 \cdot (92 : 1) - 12 \cdot (93 : 1) + 12 \cdot (94 : 1) - 52 \cdot (95 : 1), \end{aligned}$$

$$\begin{aligned} f_3 = & 50 \cdot (44 : 1) + 142 \cdot (70 : 1) - 9 \cdot (78 : 1) - 124 \cdot (79 : 1) + 34 \cdot (83 : 1) \\ & + 18 \cdot (87 : 1) - 68 \cdot (88 : 1) + 105 \cdot (89 : 1) - 34 \cdot (90 : 1) - 37 \cdot (91 : 1) \\ & - 9 \cdot (92 : 1) + 9 \cdot (93 : 1) - 9 \cdot (94 : 1) + 130 \cdot (95 : 1). \end{aligned}$$

If A_3 happens to be a Jacobian variety J_C of some curve C , the basis $\{f_1, f_2, f_3\}$ should give a basis of regular differential forms on the curve C . It turns out that this is indeed the case.

For $p = 16529$, T_p is represented by the matrix

$$\begin{pmatrix} -36 & 68 & -1 & 67/2 & -55 & 53/2 & 34 \\ 138 & 120 & 29 & 9 & 3 & 20 & -50 \\ -136 & -305 & -185 & -321 & -85 & -322 & -162 \\ 0 & 0 & 0 & 114 & 0 & 17 & 53 \\ -110 & 85 & 82 & 93 & 145 & 95 & 34 \\ 0 & 0 & 0 & 19 & 0 & 171 & -17 \\ 0 & 0 & 0 & 17 & 0 & -2 & 167 \end{pmatrix}$$

with respect to the basis $\{g_1, g_2, \dots, g_7\}$.

Using the Eichler-Shimura relation, one computes from this the characteristic polynomial f_0 of Frobenius σ_p at p (on l^n torsion or on a Tate module of A_3):

$$\begin{aligned} f_0 = & (x^6 - 452x^5 + 115418x^4 - 17978899x^3 + 1907744122x^2 - 123489944132x \\ & + 4515852403889) \cdot (x^8 - 44x^7 + 31601x^6 - 1865601x^5 + 749060774x^4 - \\ & 30836518929x^3 + 8633640983441x^2 - 198697505771116x + 74642524383881281) \end{aligned}$$

So, the characteristic polynomial f of σ_p over K_3 is the irreducible factor of f_0 of sixth degree:

$$x^6 - 452x^5 + 115418x^4 - 17978899x^3 + 1907744122x^2 - 123489944132x + 4515852403889.$$

The number h_0 of points of A_3 over \mathbb{F}_p is obtained by substituting $x = 1$ for f :

$$h_0 = f(1) = 4394252339947.$$

Since the characteristic polynomial of the fifth power of Frobenius is also easily calculated from T_p , the number h of points over a degree five extension \mathbb{F}_p^5 is obtained similarly:

$$h = 4394252339947 \times 427379515481622744216694600721926448140291414819361.$$

It is immediately verified that $q = p^5$ and h in the above satisfies the security conditions 1, 2, 3, and 4.

3.2 Defining Equation of a Simple Factor of a Modular Curve

Using the method of [13], we determine whether or not a given simple factor of $J_0(N)$ is a Jacobian J_C of some algebraic curve, and if it is, we find a defining equation of the corresponding curve C .

First, we give Shimura's result for hyperelliptic modular curves.

Algorithm 1 (Defining Equation for Hyperelliptic Modular Curves[13])

Input: a basis $\{f_1(z), f_2(z), \dots, f_g(z)\}$ for $S_2(N)$ of a hyperelliptic level N
Output: a defining polynomial $y^2 - x^{2g+2} - a_1x^{2g+1} - \dots - a_{2g+2}$

1° Calculate a Fourier expansion of every cusp form $f_i(z)$, and normalize them into the following manner:

$$\begin{aligned} f_1(z) &= q^g + s_{1,g+1}q^{g+1} + \dots + s_{1,g+i}q^{g+i} + \dots \\ f_2(z) &= q^{g-1} + s_{2,g}q^g + \dots + s_{2,g+i}q^{g+i} + \dots \\ &\dots \\ f_g(z) &= q + s_{g,2}q^2 + \dots + s_{g,g+i}q^{g+i} + \dots \end{aligned} \quad (5)$$

2° We only need Fourier coefficients of at most $(3g+3)$ -th degree.

$$x \leftarrow \frac{f_2}{f_1}, y \leftarrow \frac{q}{f_1} \frac{dx}{dq}$$

3° Calculate coefficients a_1, a_2, \dots recursively as follows:

$$\begin{aligned} y^2 - x^{2g+2} &= a_1q^{-2g-1} + \dots \\ y^2 - x^{2g+2} - a_1x^{2g+1} &= a_2q^{-2g} + \dots \\ &\dots \end{aligned}$$

The principle of Algorithm 1 is as follows. When a modular curve $X_0(N)$ is hyperelliptic, the cusp $\infty i \in \mathbb{H}$ is not a Weierstrass point. So, $X_0(N)$ has an affine model with the cusp ∞i as one of the two points at infinity:

$$y^2 = x^{2g+2} + a_1x^{2g+1} + \dots + a_{2g+2}.$$

Let α be one of the roots of the right-hand side, then a basis of regular differential forms on $X_0(N)$ is given by

$$\left\{ \frac{dx}{y}, \frac{(x-\alpha)dx}{y}, \dots, \frac{(x-\alpha)^{g-1}dx}{y} \right\}.$$

On the other hand, a basis of regular differential forms on $X_0(N)$ is also given by

$$\{f_1dz, f_2dz, \dots, f_gdz\}.$$

Therefore, we can suppose

$$x = \frac{f_2}{f_1}, y = \frac{q}{f_1} \frac{dx}{dq}.$$

For a hyperelliptic curve C obtained as a factor of a modular curve, the cusp $\infty i \in \mathbb{H}$ may be its Weierstrass point. (For example, the hyperelliptic curve of

genus 2 obtained as a factor of $X_0(68)$.) But, also in this case, C has an affine model with the cusp ∞i as a unique point at infinity:

$$y^2 = x^{2g+1} + a_1x^{2g+1} + \cdots + a_{2g+1}.$$

Putting one of the roots of the right-hand side as α , a basis of regular differential forms on C is given by

$$\left\{ \frac{dx}{y}, \frac{(x-\alpha)dx}{y}, \dots, \frac{(x-\alpha)^{g-1}dx}{y} \right\}.$$

On the other hand, regular differential forms is also spanned by

$$\{f_1dz, f_2dz, \dots, f_gdz\},$$

where $\{f_1, \dots, f_g\}$ is a basis of \mathbb{T} -submodule of $S_2(N)$ corresponding to C . Therefore, Normalizing $\{f_1, \dots, f_g\}$ as

$$\begin{aligned} f_1(z) &= q^{2g-1} + s_{1,g+1}q^{2g-2} + \cdots \\ f_2(z) &= q^{2g-3} + s_{2,g}q^{2g-4} + \cdots \\ &\dots \\ f_g(z) &= q + s_{g,2} + \cdots, \end{aligned}$$

we can also suppose

$$x = \frac{f_2}{f_1}, y = \frac{q}{f_1} \frac{dx}{dq}.$$

Note when the cusp ∞i is a Weierstrass point, x has a pole of order two at a point at infinity.

Thus, we get

Algorithm 2 (Modular Kyperelliptic; Cusp ∞i as a Weierstrass Point)

Input: a basis $\{f_1(z), f_2(z), \dots, f_g(z)\}$ of a \mathbb{T} -submodule of $S_2(N)$

Output: a defining polynomial $y^2 - 4x^{2g+1} - a_1x^{2g} - \cdots - a_{2g+1}$

1° Calculate a Fourier expansion of every cusp form $f_i(z)$, and normalize them into the following manner:

$$\begin{aligned} f_1(z) &= q^{2g-1} + s_{1,g+1}q^{2g-2} + \cdots \\ f_2(z) &= q^{2g-3} + s_{2,g}q^{2g-4} + \cdots \\ &\dots \\ f_g(z) &= q + s_{g,2} + \cdots. \end{aligned}$$

2°

$$x \leftarrow \frac{f_2}{f_1}, y \leftarrow \frac{q}{f_1} \frac{dx}{dq}$$

3° Calculate coefficients a_1, a_2, \dots recursively as follows:

$$\begin{aligned} y^2 - 4x^{2g+1} &= a_1 q^{-4g} + \dots \\ y^2 - 4x^{2g+1} - a_1 x^{2g} &= a_2 q^{-4g+2} + \dots \\ &\dots \end{aligned}$$

In general, for an algebraic curve C which is not hyperelliptic, letting a basis of space of regular differential forms $H^0(\Omega_C^1)$ be $\{\omega_1, \dots, \omega_g\}$, the map

$$\begin{aligned} \Phi : C &\longrightarrow \mathbb{P}^{g-1} \\ P &\mapsto \left(1, \frac{\omega_2}{\omega_1}(P), \dots, \frac{\omega_g}{\omega_1}(P)\right) \end{aligned}$$

is an embedding morphism, and its image $\text{Im}(\Phi)$ is a nonsingular algebraic curve in \mathbb{P}^{g-1} , called a “canonical curve” of C .

In the case of modular curve or its factor, its canonical curve is just an algebraic curve in \mathbb{P}^{g-1} defined by the relations among $\{f_1, \dots, f_g\}$, which is a basis of the corresponding \mathbb{T} -submodule of $S_2(N)$.

A canonical curve of genus three is a plane quartic curve. As Shimura pointed out [13], the following Theorem 2 is useful for a canonical curve of genus four or more.

Theorem 2 (Petri's Theorem [1]). *Let C be a canonical curve of genus four or more. Then C is an intersection of some quadratic hypersurfaces, or an intersection of some quadratic and cubic hypersurfaces.*

By Theorem 2, for a curve C obtained as a factor of a modular curve, we only need to find quadratic or cubic relations among f_1, \dots, f_g in order to obtain a canonical curve of C . Shimura estimates the number of relations as in Table 1 [13].

Table 1. Number of equations for canonical curves

genus	equations
3	one quartic relation
4	one quadratic and one cubic relations
5	three quadratic relations
...	...

Each explicit relation is obtained easily using the Fourier expansions of a basis $\{f_1(z), f_2(z), \dots, f_g(z)\}$.

Take an abelian variety A obtained as a simple factor of $J_0(N)$. Let K be a \mathbb{T} -submodule of $S_2(N)$ corresponding to A , and $\{f_1, \dots, f_g\}$ be its basis over \mathbb{Q} .

Now, we can determine whether A is a Jacobian J_C of some algebraic curve or not, and if it is, we can find a defining equation of the corresponding curve C , as follows:

Algorithm 3 (Defining Equation of Simple Factor of a Modular Curve)

Input: a basis $\{f_1, \dots, f_g\}$ of a \mathbb{T} -submodule of $S_2(N)$ over \mathbb{Q} , corresponding to a simple factor A of $J_0(N)$

Output: ‘null’ or a defining polynomial F of an algebraic curve C with Jacobian $J_C \simeq A$

- 1° Calculate a Fourier expansion of every cusp form $f_i(z)$, and determine the cusp ∞i is a Weierstrass point or not. That the cusp ∞i is not a Weierstrass point is equivalent to the fact that $\{f_1, \dots, f_g\}$ are expanded just as in Equation (5) in Algorithm 1.
- 2° Assume A is a Jacobian of some hyperelliptic curve. Calculate a defining polynomial $F(x, y)$ of the hyperelliptic curve, using Algorithm 1 when the cusp ∞i is not a Weierstrass point, or Algorithm 2 when the cusp ∞i is a Weierstrass point.
- 3° Check the validity of the polynomial $F(x, y)$. That is, substitute $x = \frac{f_2}{f_1}$, and $y = \frac{q}{f_1} \frac{dx}{dq}$ for $F(x, y)$, and see whether the resulting Fourier coefficients vanish. If it is, output $F(x, y)$ and terminate.
- 4° Assume C is a Jacobian of some non-hyperelliptic curve C , and calculate the canonical curve of C . That is, find all the quadratic or cubic relations F among $\{f_1, \dots, f_g\}$. And see whether the curve defined by F is nonsingular. If it is, output F and terminate. Else A is supposed to be not a Jacobian variety, and output ‘null’.

Algorithm 3 is not strict with mathematics, of course. It is not proved that the output of Algorithm 3 defines an algebraic curve with Jacobian A . However, remember that our aim is to construct a secure C_{ab} curve. It is sufficient that the resulting curve has a Jacobian of the expected order in fact.

Example: Level 97. In section 3.1, we guessed that Jacobian $J_0(97)$ has a three-dimensional simple factor A_3 . Also, we obtained the basis $\{f_1, f_2, f_3\}$ of the corresponding \mathbb{T} -submodule K_3 of $S_2(N)$. Here, we perform Algorithm 3 with the basis $\{f_1, f_2, f_3\}$ as an input.

- 1° Calculating and normalizing Fourier expansions of $\{f_1, f_2, f_3\}$, we get

$$\begin{aligned} f_1 &= q^3 - q^4 - 2q^5 - q^6 + q^7 + 4q^8 - 2q^9 + 3q^{10} + q^{12} - q^{14} - 7q^{16} - q^{17} + q^{18} \dots \\ f_2 &= q^2 - 3q^4 - q^5 - 2q^6 + 5q^8 + q^9 + 2q^{10} + q^{11} + 5q^{12} - q^{13} - 3q^{14} - 8q^{16} + \dots \\ f_3 &= q - 4q^4 - 5q^5 - 3q^6 - q^7 + 9q^8 - q^9 + 8q^{10} - q^{11} + 7q^{12} - 2q^{13} - 3q^{14} - \dots \end{aligned}$$

Coefficients are calculated up to 80-th degree. From this, we know the cusp ∞i is not a Weierstrass point.

- 2° Assuming A_3 is an Jacobian of some hyperelliptic curve, we calculate a defining equation of the hyperelliptic curve, using Algorithm 1.

1°° Fourier expansions of $\{f_1, f_2, f_3\}$ was already computed at 1°.

2^{oo} We obtain

$$\begin{aligned} x &= \frac{f_2}{f_1} \\ &= 1 + q^{-1} + 2q^2 - q^4 + 4q^5 - 2q^7 + 7q^8 + q^9 - 5q^{10} + 13q^{11} - 9q^{13} + \dots \\ y &= \frac{q}{f_1} \frac{dx}{dq} \\ &= -8 - q^{-4} - q^{-3} - 3/q^2 - 2/q - 14q - 7q^2 - 28q^3 - 57q^4 - \dots \end{aligned} \quad (6)$$

Actually, coefficients are calculated up to 75-th degree.

3^{oo} We obtain the defining polynomial

$$-23 + 182x - 241x^2 + 210x^3 - 136x^4 + 62x^5 - 21x^6 + 6x^7 - x^8 + y^2. \quad (7)$$

3° Substituting Equation (6) for Polynomial (7), we encounter

$$70q + 14q^2 - 300q^3 + 398q^4 + 174q^5 - 1106q^6 + 930q^7 + 479q^8 - 472q^9 - 1572q^{10} - \dots$$

As coefficients does not vanish, we determine A_3 is not a Jacobian of any hyperelliptic curve.

4° Assuming A_3 is an Jacobian of some non-hyperelliptic curve C , we calculate the canonical curve of C . As the genus of C is three, the defining polynomial is a single quartic equation F among $Z = f_1, Y = f_2, X = f_3$. Using the above Fourier expansion of X, Y, Z , we obtain the unique relation

$$F = -2X^4 - X^3Y - 3X^2Y^2 + 6X^3Z + 3X^2YZ + XY^2Z + Y^3Z - 5X^2Z^2 - Y^2Z^2 + XZ^3.$$

As $F = 0$ defines a nonsingular curve, we determine the simple factor A_3 is a Jacobian of the curve $F = 0$.

3.3 C_{ab} Model of a Simple Factor of a Modular Curve

In the last section, we got an explicit defining equation of a simple factor of a modular curve. Here we translate it into a C_{ab} curve.

In the hyperelliptic case, we obtain the equation of the form

$$y^2 = x^{2g+2} + a_1x^{2g+1} + \dots + a_{2g+2},$$

besides C_{ab} curve of type $\{2, 2g+1\}$. Factoring the right-hand side, we get

$$y^2 = (x + \lambda_1)(x + \lambda_2) \cdots (x + \lambda_{2g+2}),$$

and dividing two sides by $(x + \lambda)^{2g+2}$, we get

$$\frac{y^2}{(x + \lambda)^{2g+2}} = 1 \cdot \prod_{i=2}^{2g+2} \left(1 + \frac{\lambda_i - \lambda_1}{x + \lambda_1}\right).$$

So, putting

$$X = \frac{1}{x + \lambda_1}, Y = \frac{y}{(x + \lambda_1)^{g+1}},$$

we get

$$Y^2 = \prod_{i=2}^{2g+2} (1 + (\lambda_i - \lambda_1)X)$$

This is a C_{ab} curve of type $\{2, 2g + 1\}$.

We now consider the non-hyperelliptic simple factor, we obtained its canonical curve C . By Theorem 1, in order to translate C into a C_{ab} curve, we only need to find the generator $A = \langle a_1, a_2, \dots, a_t \rangle$ of the monoid M_Q and to find the function $f_i \in L(\infty Q)$ ($i = 1, 2, \dots, t$) with pole order i at Q , for some rational point Q on C . Therefore, all we have to do is to find a basis $L(mQ)$ for some rational point Q on C ($3 \leq m \leq a_t$).

As the canonical curve C is nonsingular, it is not difficult to find a basis of $L(D)$ for any divisor D [8]:

Proposition 2. *Let $D = \sum n_i P_i - \sum m_j Q_j$ be a divisor on a nonsingular curve C with non-negative integers n_i, m_j . Let $I_1 = \cap I_{P_i}^{n_i}, I_2 = \cap I_{Q_j}^{m_j}$, where I_P is the maximal ideal corresponding to P of the coordinate ring R of C . Fix $(0 \neq) \forall f \in I_1$.*

Then, we have

$$L(D) = \left\{ \frac{g}{f} \mid g \in fI_2 : I_1 \right\}.$$

Proof. As C is nonsingular, any element in $L(D)$ can be written as $\frac{g}{f}$ for some $g \in R$. Then,

$$\begin{aligned} \frac{g}{f} &\in L(D) \\ \Leftrightarrow \frac{g}{f} I_1 &\subset I_2 \\ \Leftrightarrow g &\in fI_2 : I_1 \end{aligned}$$

□

The number of equations for a C_{ab} curve becomes the smallest when we take a Weierstrass point as the base point. So, it is desirable to choose a Weierstrass point as the point Q in the above. When the genus is three, Weierstrass points of a canonical curve are easily found.

Let C be a canonical curve of genus three. Let K be a canonical series of C . As $\dim(K) = 2$, that a point Q is a Weierstrass point is equivalent to the fact that the tangent line at the point Q meets the curve C with the multiplicity three or more. So, a Weierstrass point Q is a common zero of

$$\begin{aligned} f(x, y) &= 0 \\ D_{a,b}f(x, y) &= 0 \\ D_{a,b}^{(2)}f(x, y) &= 0, \end{aligned}$$

where

$$D_{a,b}f(x, y) = a\partial f/\partial x + b\partial f/\partial y,$$

and $D_{a,b}^{(2)}f(x, y) = D_{a,b}(D_{a,b}(f))$.

Example: Level 97. We saw that the Jacobian $J_0(97)$ of the modular curve $X_0(97)$ has a simple factor A_3 of dimension three, and that A_3 is a Jacobian of an algebraic curve C (ref. section 3.2). The defining equation of the canonical curve of C was given by

$$f = -2x^4 - x^3y - 3x^2y^2 + 6x^3 + 3x^2y + xy^2 + y^3 - 5x^2 - y^2 + x.$$

For $p = 16529$, we find a Weierstrass point of the curve $f = 0$ over \mathbb{F}_p . As equations

$$\begin{aligned} f(x, y) &= x - 5x^2 + 6x^3 - 2x^4 + 3x^2y - x^3y - y^2 + xy^2 - 3x^2y^2 + y^3 \\ D_{a,b}f(x, y) &= a - 10ax + 3x^2 + 18ax^2 - x^3 - 8ax^3 - 2y + 2xy + 6axy - 6x^2y \\ &\quad - 3ax^2y + 3y^2 + ay^2 - 6axy^2 \\ D_{a,b}^{(2)}f(x, y) &= -2 - 10a^2 + 2x + 12ax + 36a^2x - 6x^2 - 6ax^2 - 24a^2x^2 + 6y \\ &\quad + 4ay + 6a^2y - 24axy - 6a^2xy - 6a^2y^2 \end{aligned}$$

has a common zero

$$a = 12900, x = 13695, y = 14705,$$

$Q = (13695, 14705)$ is a Weierstrass point.

Calculating $l(m) = \dim L(mQ)$ ($m = 3, 4, 5, 6, 7$) by Prop. 2, we have

$$l(3) = 2, l(4) = 2, l(5) = 3, l(6) = 4, l(7) = 5.$$

So, we know gap sequences at the point Q is 1,2,4. Hence,

$$M_Q = \langle 3, 5, 7 \rangle.$$

Thus, the curve C is a $C_{3,5,7}$ curve.

The function $X \in L(3Q)$ with $v_Q(X) = -3$, the function $Y \in L(4Q)$ with $v_Q(Y) = -4$, and the function $Z \in L(5Q)$ with $v_Q(Z) = -5$ are given by

$$\begin{aligned} X &= (12855 + 11167x + 5996x^2 + x^3 + 9720y + 10529xy + 4636x^2y + 10496y^2 \\ &\quad + 10744xy^2)/(13280 + 13941y + 5472y^2 + y^3) \\ Y &= (8608 + 6182x + 8423x^2 + 15577x^3 + 13719y + 7604xy + 424x^2y + 8263x^3y \\ &\quad + 7442y^2 + 9157xy^2 + 7894x^2y^2 + 4131x^3y^2 + 14194y^3 + 12726xy^3 \\ &\quad + 9702x^2y^3 + 15348y^4 + 5202xy^4)/(9403 + 5617y + 568y^2 + 13412y^3 \\ &\quad + 9120y^4 + y^5) \\ Z &= (10644 + 13291x + 7571x^2 + 8617x^3 + 2836y + 15714xy + 1350x^2y + x^3y \\ &\quad + 667y^2 + 3987xy^2 + 11840x^2y^2 + 2036x^3y^2 + 1947y^3 + 1150xy^3 \\ &\quad + 12002x^2y^3 + 6207x^3y^3 + 15337y^4 + 7047xy^4 + 8184x^2y^4 + 13431x^3y^4 \\ &\quad + 8564y^5 + 5258xy^5 + 14541x^2y^5 + 9149y^6 + 7639xy^6)/(9594 + 7377y \\ &\quad + 15644y^2 + 6261y^3 + 1988y^4 + 14942y^5 + 12768y^6 + y^7). \end{aligned} \tag{8}$$

A general form of defining equations of $C_{3,5,7}$ curve is Equation (2). In this case, we have

$$\begin{aligned} 0 &= 11654X + 6133X^2 + 10293X^3 + 3017Y + 463XY + Y^2 + 7669Z + 15127XZ \\ 0 &= 15687X + 8029X^2 + 10416X^3 + 9882X^4 + 14252Y + 6982XY + 9150X^2Y \\ &\quad + 4600Z + 6150XZ + YZ \\ 0 &= 1362X + 11237X^2 + 3867X^3 + 95X^4 + 8346Y + 9761XY + 10084X^2Y \\ &\quad + 5949X^3Y + 1677Z + 7169XZ + 831X^2Z + Z^2. \end{aligned}$$

By the result of section 3.1, we guess that the above $C_{3,5,7}$ curve has a Jacobian of the order

$$h = 4394252339947 \times 427379515481622744216694600721926448140291414819361$$

over the finite field \mathbb{F}_{p^5} for $p = 16529$. In fact, it is verified that h times a random rational point of the curve over \mathbb{F}_{p^5} is equal to the unit element of the Jacobian, using the addition algorithm in [2].

Similarly, secure examples in genus 2 and 3 with C_{ab} equations were found for 21 different levels $N \leq 109$.

Acknowledgments

I wish to thank anonymous referees for their a lot of efforts to help me arrive at a final version of the paper.

References

1. E. Arbarello, M.Cornalba, P.A.Griffiths, and J.Harris, “Geometry of Algebraic Curves Volume I,” Springer-Verlag, 1984.
2. S. Arita, “Algorithms for computations in Jacobian group of C_{ab} curve and their application to discrete-log-based public key cryptosystems,” Conference on The Mathematics of Public Key Cryptography, Toronto, 1999.
3. S. Arita, “Gaudry’s variant against C_{ab} curve,” LNCS 1751, Proceedings of PKC 2000, pp. 58-67
4. J.E.Cremona, “Algorithms For Modular Elliptic Curves”, Cambridge University Press, 1997.
5. G.Frey and H.-G.Rück, “A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves”, Mathematics of Computation, 62 (1994), 865-874.
6. G.Frey and M. Müller, “Arithmetic of Modular Curves and Applications”, preprint, 1998.
7. P.Gaudry, “A variant of the Adleman-DeMarris-Huang algorithm and its application to small genera,” Conference on The Mathematics of Public Key Cryptography, Toronto, 1999.
8. D.Grayson, M.Stillman, “Macaulay 2 – a system for computation in algebraic geometry and commutative algebra”, <http://math.uiuc.edu/Macaulay2>.
9. S. Miura, “Linear Codes on Affine Algebraic Curves”, Trans. of IEICE, vol. J81-A, No. 10, 1398-1421, Oct. 1998.

10. J.M.Pollard, "Monte Carlo methods for index computation mod p," *Math. Comp.*,32(143),pp.918-924,1978.
11. G.Cornell, J.H.Silverman, G.Stevens (ed), "Modular Forms and Fermat's Last Theorem", Springer, 1997.
12. H.-G.Rück, "On the discrete logarithm in the divisor class group of curves," *Math. Comp.*,68(226),pp.805-806,1999.
13. M. Shimura, "Defining Equations of Modular Curves $X_0(N)$ ", *Tokyo J. Math.*, Vol. 18, No. 2 (1995), pp.443-456.
14. X.Wang, "2-dimensional simple factors of $J_0(N)$ ", *Manuscripta Math.* 87 (1995), pp. 179-197.
15. H-J. Weber, "Hyperelliptic Simple Factors of $J_0(N)$ with Dimension at Least 3", *Experimental Mathematics* 6:4 (1997), pp. 273-287.

Curves over Finite Fields with Many Rational Points Obtained by Ray Class Field Extensions

Roland Auer

Rijksuniversiteit Groningen, Vakgroep Wiskunde
Blauwborgje 3, NL-9747 AC Groningen, The Netherlands
auer@math.rug.nl

Abstract. A general type of ray class fields of global function fields is investigated. The computation of their genera is reduced to the determination of the degrees of these extensions, which turns out to be the main difficulty. While in two special situations explicit formulas for the degrees are known, the general problem is solved algorithmically. The systematic application of the methods described yields several new examples of algebraic curves over \mathbb{F}_2 , \mathbb{F}_3 , \mathbb{F}_4 , \mathbb{F}_5 and \mathbb{F}_7 with comparatively many rational points.

1 Introduction

The maximum number of \mathbb{F}_q -rational points on a (smooth, projective, absolutely irreducible algebraic) curve $X|\mathbb{F}_q$ of genus $g(X) = g$ defined over the finite field \mathbb{F}_q is usually denoted by $N_q(g)$. In the early eighties, Serre [20,21,22] has written down formulas for $N_q(1)$ and $N_q(2)$.

Since the precise value of $N_q(g)$ is quite difficult to determine in general, the work of many mathematicians has instead led to large tables, such as [6], giving an interval for this quantity. The lower bounds are usually realized by Abelian coverings of small genus curves, which are either given by explicit equations (Hansen and Stichtenoth [7], [8], [24], van der Geer and van der Vlugt [3], [4], [5], Niederreiter and Xing [12], [13], [15], Shabat [23], and others) or obtained by class field theory or an equivalent construction (Schoof [19], Lauter [11], Niederreiter and Xing [25], [13], [14], [16]). (Please note that these references are far from being complete.) The present paper, which adds to the second category, summarizes the author's thesis [1], where all results stated here are proved in detail.

Since we employ ray class field extensions, to the curve $X|\mathbb{F}_q$ we associate the global function field $K = \mathbb{F}_q(X)$. Its genus g_K equals $g(X)$, and coverings of X correspond to field extensions of K , the degree of the covering being the degree of the extension. By construction, \mathbb{F}_q is the full constant field, i.e. is algebraically closed in K , and we express this instance by writing $K|\mathbb{F}_q$.

A place of K , by which we mean the maximal ideal \mathfrak{p} in some discrete valuation ring of K , with (residue field) degree $d = \deg \mathfrak{p}$, corresponds to (a Galois

conjugacy class of) d points in $X(\mathbb{F}_{q^d})$, and each point on X having \mathbb{F}_{q^d} as its minimal field of definition over \mathbb{F}_q lies in such a conjugacy class. In particular $K|\mathbb{F}_q$ has $N_K = |X(\mathbb{F}_q)|$ rational places, i.e. places of degree 1. Throughout this paper, $K|\mathbb{F}_q$ is a global function field, and all algebraic extensions of K are assumed to lie in some fixed algebraic closure \bar{K} of K

2 Ray Class Fields

We fix a non-empty set S of places of K , and denote the greatest common divisor of the degrees of its elements by $d := \gcd\{\deg \mathfrak{p} \mid \mathfrak{p} \in S\}$. Let \mathfrak{m} be an S -cycle, i.e. an effective divisor of K with support away from S . We consider the S -ray class field mod \mathfrak{m} , denoted $K_S^\mathfrak{m}$, which is defined as the largest Abelian extension $L|K$ of conductor at most \mathfrak{m} such that every place of S splits completely in L . These extensions occur e.g. in Perret [17].

In the special case of $\mathfrak{m} = \mathfrak{o}$ (the zero element in the divisor group), $K_S^\mathfrak{o}$ is also known as the S -Hilbert class field (cf. Rosen [18]). We recall that the Galois group $G(K_S^\mathfrak{o}|K)$ is isomorphic to the (ideal) class group $\mathcal{C}\ell(\mathcal{O}_S)$ of the Dedekind ring \mathcal{O}_S consisting of all functions with poles only in S .

Since S is non-empty, by class field theory $K_S^\mathfrak{m}$ is a finite (algebraic) extension of K . In fact, using Čebotarev's Density Theorem, any finite Abelian $L|K$ is seen to be equal to some $K_S^\mathfrak{m}$. Here for \mathfrak{m} we can take the conductor of $L|K$, and S can always be chosen finite. Furthermore, the ray class fields satisfy the following properties.

Proposition 1. *Let S , d and \mathfrak{m} be as above, T another non-empty set of places of K and \mathfrak{n} a T -cycle.*

- (a) *The full constant field of $K_S^\mathfrak{m}$ has degree d over \mathbb{F}_q , thus $K_S^\mathfrak{m}|\mathbb{F}_{q^d}$.*
- (b) *If $S \supseteq T$ and $\mathfrak{m} \leq \mathfrak{n}$, then $K_S^\mathfrak{m} \subseteq K_T^\mathfrak{n}$.*
- (c) *$K_S^\mathfrak{m} \cap K_T^\mathfrak{n} = K_{S \cup T}^{\min\{\mathfrak{m}, \mathfrak{n}\}}$, where the minimum is taken coefficient wise.*

In terms of the ray class fields of K , we can write down the genus for any Abelian extension of K .

Theorem 1. *Let S , d , $\mathfrak{m} = \sum_p m_{\mathfrak{p}} \mathfrak{p}$ be as above, and L an intermediate field of $K_S^\mathfrak{m}|K$. Then the genera g_K and g_L of K and L satisfy*

$$d \cdot (g_L - 1) = [L : K] \left(g_K - 1 + \frac{\deg \mathfrak{m}}{2} \right) - \frac{1}{2} \sum_{\mathfrak{p}} \sum_{n=1}^{m_{\mathfrak{p}}} [L \cap K_S^{\mathfrak{m}-n\mathfrak{p}} : K] \deg \mathfrak{p} .$$

This formula can be proved either by applying Möbius inversion to the Conductor Discriminant Product Formula, as done by Cohen et al. [2] in the number field case, or by using Hilbert's Different Formula, the Hasse-Arf Theorem and the connection between upper ramification groups and higher unit groups known from local class field theory (see [1]).

We observe that computing the genus of ray class field extensions amounts to determining their degrees. This is easily done if S consists of just one place, but becomes much more intricate if we require more places to split. In the following section we indicate an algorithmic solution of this problem.

3 Computation of Degrees

For simplicity we shall restrict to the case of ramification at only one place \mathfrak{p} of $K|\mathbb{F}_q$ (outside S), i.e. to $\mathfrak{m} = m\mathfrak{p}$ with $m \in \mathbb{N}_0$. Recall that, by definition, the residue field $\mathbb{F}_{\mathfrak{p}}$ of \mathfrak{p} satisfies $[\mathbb{F}_{\mathfrak{p}} : \mathbb{F}_q] = \deg \mathfrak{p}$. We determine the degrees $[K_S^{m\mathfrak{p}} : K]$ in three steps.

First of all, from what has been said about the Hilbert class field, $[K_S^{\mathfrak{o}} : K]$ equals the S -class number $h_S := |\mathcal{C}(\mathcal{O}_S)|$. Its computation is connected with the problem of finding generators for the group \mathcal{O}_S^* of S -units, which in turn are needed for the other two steps.

Indeed, by class field theory, the Galois group $G(K_S^{\mathfrak{p}}|K_S^{\mathfrak{o}})$ is isomorphic to the cokernel of the canonical group homomorphism $\mathcal{O}_S^* \rightarrow \mathbb{F}_{\mathfrak{p}}^*$. Since $\mathbb{F}_{\mathfrak{p}}^* \subseteq \mathcal{O}_S^*$, it follows that $[K_S^{\mathfrak{p}} : K_S^{\mathfrak{o}}]$ divides $\frac{q^{\deg \mathfrak{p}} - 1}{q - 1}$.

Similarly, the cokernel of $\mathcal{O}_S^* \cap (1 + \mathfrak{p}) \rightarrow (1 + \mathfrak{p})/(1 + \mathfrak{p}^m)$ is isomorphic to $G(K_S^{m\mathfrak{p}}|K_S^{\mathfrak{p}})$. According to the following theorem, its order can be determined for all $m \in \mathbb{N}$ simultaneously. Let p be the characteristic of K , and define

$$\lceil a \rceil_p := \min\{p^l \mid a \leq p^l, l \in \mathbb{N}_0\}$$

for any real $a > 0$.

Theorem 2. *There are $s := |S| - 1$ positive integers n_1, \dots, n_s depending only on S and \mathfrak{p} such that*

$$[K_S^{m\mathfrak{p}} : K_S^{\mathfrak{p}}] = q^{(m-1)\deg \mathfrak{p}} \Big/ \prod_{i=1}^s \left\lceil \frac{m}{n_i} \right\rceil_p$$

for all $m \in \mathbb{N}$.

The proof given in [1, p. 43] provides an algorithm for the computation of the numbers n_1, \dots, n_s . Since the order of these numbers is irrelevant, the behavior of the degrees can be summarized in the polynomial

$$\delta_{S,\mathfrak{p}} := \sum_{i=1}^s t^{n_i},$$

which is uniquely determined by S and \mathfrak{p} . Unfortunately we have no explicit formula for $\delta_{S,\mathfrak{p}}$ except in some particular cases, which are treated in the next section.

4 Rational Function Field

Here we want to draw the attention to two special situations where the polynomial $\delta_{S,\mathfrak{p}}$ can be given explicitly.

Theorem 3. *Let K be the rational function field over the prime field \mathbb{F}_p , S a non-empty set of rational places of K and \mathfrak{p} a rational place of K not occurring in S ; thus $0 \leq s := |S| - 1 < p$. Then $\delta_{S,\mathfrak{p}} = \sum_{n=1}^s t^n$.*

In particular $K_S^{(s+1)\mathfrak{p}} = K$, and for $m \geq s + 2$ it follows that $K_S^{m\mathfrak{p}}$ has exactly

$$N_{K_S^{m\mathfrak{p}}} = 1 + [K_S^{m\mathfrak{p}} : K](s + 1)$$

rational places. The genus can be calculated by means of Theorem 1. As an example we have carried out these computations for $p \in \{5, 7\}$ and different values of s and m . The results are displayed in the tables 1 and 2.

Table 1. Ray class fields over \mathbb{F}_5

s	1	2	3	4	1	2	3	4	1	2	3	4
m	3	4	5-6	7	4	5-6	7	8	5-6	7	8	9
$[K_S^{m\mathfrak{p}} : K]$	5	5	5	5	25	25	25	25	125	125	125	125
$g_{K_S^{m\mathfrak{p}}}$	2	4	6	10	22	34	56	70	172	284	356	420
$N_{K_S^{m\mathfrak{p}}}$	11	16	21	26	51	76	101	126	251	376	501	626

Table 2. Ray class fields over \mathbb{F}_7

s	1	2	3	4	5	6	1	2	3	4	5	6
m	3	4	5	6	7-8	9	4	5	6	7-8	9	10
$[K_S^{m\mathfrak{p}} : K]$	7	7	7	7	7	49	49	49	49	49	49	49
$g_{K_S^{m\mathfrak{p}}}$	3	6	9	12	15	21	45	69	93	117	162	189
$N_{K_S^{m\mathfrak{p}}}$	15	22	29	36	43	50	99	148	197	246	295	344

Now let $q = p^e$ with $e \in \mathbb{N}$ be an arbitrary power of the characteristic p again. Take $Q := \{1, \dots, q - 1\} \subseteq \mathbb{Z}$ as a set of representatives for the cyclic group $\mathbb{Z}/(q - 1)\mathbb{Z} \simeq \mathbb{F}_q^*$. Via this latter isomorphism, the group $G := G(\mathbb{F}_q|\mathbb{F}_p)$ acts on Q . Clearly, two elements $n, n' \in Q$ lie in the same G -orbit $Gn = Gn'$ iff $n' \equiv p^l n \pmod{q - 1}$ for some $l \in \mathbb{N}_0$. For $n \in \mathbb{N}$ we define

$$e_n := \begin{cases} |Gn| & \text{if } n \in Q \text{ and } n = \min Gn, \\ 0 & \text{otherwise.} \end{cases}$$

The following has been proved by Lauter [11].

Theorem 4. *Let K be the rational function field over \mathbb{F}_q , \mathfrak{p} a rational place of K , and S the set consisting of the other q rational places. Then $\delta_{S,\mathfrak{p}} = \sum_{n \in \mathbb{N}} e_n t^n$.*

Let us set $r := \sqrt{q}$ or \sqrt{pq} according to whether q is a square or not. By investigating the numbers e_n , we see that $K_S^{(r+1)\mathfrak{p}} = K$. Furthermore we obtain

two fields, namely $K_S^{(r+2)\mathfrak{p}}$ and $K_S^{(2r+2)\mathfrak{p}}$ of degree r and rq over K if q is a square, and $p-1$ fields $K_S^{(r+i+1)\mathfrak{p}}$ with $1 \leq i < p$ of degree q^i over K in case q is non-square. Lauter [10] has pointed out that the corresponding curves generalize certain families of Deligne-Lusztig curves. Here we want to write down defining equations for them, which might have been found by J. P. Pedersen before but without publishing.

Proposition 2. *Let $K = \mathbb{F}_q(x)$ with x an indeterminate over \mathbb{F}_q such that \mathfrak{p} is the pole of x , and let S (as in Theorem 4) consist of the remaining q rational places of K .*

- (a) *Assume that $r := \sqrt{q} \in \mathbb{N}$ and let $y, z \in \bar{K}$ satisfy $y^r + y = x^{r+1}$ and $z^q - z = x^{2r}(x^q - x)$. Then $K_S^{(r+2)\mathfrak{p}} = K(y)$ and $K_S^{(2r+2)\mathfrak{p}} = K(y, z)$.*
- (b) *For $r := \sqrt{pq} \in \mathbb{N}$ let $y_1, \dots, y_{p-1} \in \bar{K}$ satisfy $y_i^q - y_i = x^{ir/p}(x^q - x)$. Then $K_S^{(r+i+1)\mathfrak{p}} = K(y_1, \dots, y_i)$ for $i \in \{1, \dots, p-1\}$.*

5 Tables

Now we use ray class field extensions of small genus ground fields $K|\mathbb{F}_q$ to produce curves of higher genus with many rational points over \mathbb{F}_2 , \mathbb{F}_3 and \mathbb{F}_4 . Like in the tables [6] by van der Geer and van der Vlugt, we restrict ourselves to genus $g \leq 50$ and give a range (or the precise value) for $N_q(g)$. The upper bound is taken from [6]. The lower bound is attained by a field $L|\mathbb{F}_q$ of genus $g_L = g$ with precisely that many rational places, and is set in boldface if our example actually improves the lower bound known before. The field L satisfies $K_S^{(m-1)\mathfrak{p}} \subsetneq L \subseteq K_S^{m\mathfrak{p}}$ (thus has conductor $m\mathfrak{p}$) with $m \in \mathbb{N}$, \mathfrak{p} a place of K of degree $d := \deg \mathfrak{p}$, and S a non-empty set of rational places of K not containing \mathfrak{p} .

The degrees $[K_S^{m\mathfrak{p}} : K]$ are computed as indicated in Sect. 3. In order to search through a large variety of possibilities for \mathfrak{p} and S , the algorithms have been implemented in KASH/KANT [9]. Then, by Theorem 1, the genus of L is

$$g_L = 1 + [L : K] \left(g_K - 1 + \frac{md}{2} \right) - \frac{d}{2} \sum_{n=0}^{m-1} [K_S^{n\mathfrak{p}} : K] .$$

Since the inertia degree of \mathfrak{p} in L is $h_S/h_{S \cup \{\mathfrak{p}\}}$, L has

$$N_L \geq [L : K] |S| + \begin{cases} h_S & \text{if } h_S = h_{S \cup \{\mathfrak{p}\}} \text{ and } d = 1 \\ 0 & \text{otherwise} \end{cases}$$

rational places. Here equality holds iff S already contains all rational places of K that split completely in L , which in our examples is always the case. Complete information on the precise construction of each field L occurring in the tables 3–5 is given in [1].

Table 3. Function fields over \mathbb{F}_2 with many rational places

g	$N_2(g)$	$[L : K]$	$ S $	m	d	g_K	g	$N_2(g)$	$[L : K]$	$ S $	m	d	g_K
6	10	10	1	2	1	1	27	24 –25	12	2	3	2	1
7	10	2	5	2	6	1	28	25 –26	8	3	7	1	2
8	11	2	5	10	1	2	29	25–27	4	6	14	1	3
9	12	4	3	4	2	0	30	25 –27	4	6	12	1	4
10	13	4	3	4	1	2	35	29 –31	4	7	16	1	4
12	14–15	7	2	1	6	0	37	29 –32	4	7	14	1	5
14	15–16	15	1	1	4	0	39	33	16	2	8	1	0
15	17	8	2	7	1	0	41	33 –35	8	4	6	1	4
16	17 –18	2	8	14	1	5	42	33 –35	8	4	8	1	3
17	17–18	16	1	5	1	0	44	33 –37	8	4	11	1	2
19	20	4	5	2	6	1	49	36–40	12	3	1	6	3
22	21–22	4	5	12	1	2	50	40	8	5	2	7	1

Table 4. Function fields over \mathbb{F}_3 with many rational places

g	$N_3(g)$	$[L : K]$	$ S $	m	d	g_K	g	$N_3(g)$	$[L : K]$	$ S $	m	d	g_K
5	12–13	3	4	2	2	1	33	46 –49	9	5	4	1	3
7	16	8	2	1	4	0	34	45 –50	9	5	3	3	1
9	19	3	6	5	1	2	36	46–52	9	5	9	1	1
10	19–21	9	2	5	1	0	37	48–54	24	2	1	2	2
14	24–26	3	8	5	2	2	39	48 –56	24	2	2	2	1
15	28	9	3	6	1	0	43	55–60	9	6	11	1	1
16	27–29	9	3	3	2	0	45	54 –62	18	3	3	2	1
17	24–30	6	4	5	1	2	46	55–63	27	2	6	1	0
19	28–32	3	9	12	1	3	47	54 –65	18	3	6	1	1
22	30–36	3	10	3	5	3	48	55–66	9	6	11	1	2
24	31 –38	3	10	14	1	4	49	63–67	9	7	3	4	1
30	37–46	9	4	8	1	1	50	56–68	28	2	1	2	2

References

1. R. Auer. *Ray class fields of global function fields with many rational places*. Dissertation at the University of Oldenburg, 1999,
<http://www.bis.uni-oldenburg.de/dissertation/ediss.html>.
2. H. Cohen, F. Diaz y Diaz, M. Olivier. *Computing ray class groups, conductors and discriminants*. In: *Algorithmic Number Theory*. H. Cohen (ed.). Lecture Notes in Computer Science **1122**, Springer, Berlin, (1996) 51–59.
3. G. van der Geer, M. van der Vlugt. *Curves over finite fields of characteristic 2 with many rational points*. C. R. Acad. Sci. Paris **317** (1993) série I, 593–597.
4. G. van der Geer, M. van der Vlugt. *How to construct curves over finite fields with many points*. In: *Arithmetic Geometry (Cortona 1984)*. F. Catanese (ed.). Cambridge Univ. Press (1997) 169–189.

Table 5. Function fields over \mathbb{F}_4 with many rational places

g	$N_4(g)$	$[L : K]$	$ S $	m	d	g_K	g	$N_4(g)$	$[L : K]$	$ S $	m	d	g_K
4	15	2	7	6	1	1	24	49–52	4	12	10	1	3
5	17–18	4	4	6	1	0	25	51–53	12	4	3	1	2
6	20	4	5	2	3	0	27	49–56	16	3	6	1	0
8	21–24	2	10	6	1	3	28	53–58	4	13	14	1	3
9	26	8	3	3	1	1	31	60–63	15	4	1	3	2
10	27–28	12	2	2	1	1	32	57–65	8	7	10	1	1
11	26–30	2	13	4	3	3	33	65–66	16	4	7	1	0
12	29–31	4	7	8	1	1	34	57–68	8	7	8	1	2
13	33	8	4	6	1	0	36	64–71	8	8	3	3	2
14	32–35	8	4	2	3	0	41	65–78	20	3	3	1	2
19	37–43	4	9	10	1	2	43	72–81	24	3	2	3	0
20	40–45	8	5	3	3	0	45	80–84	16	5	4	2	0
21	41–47	4	10	8	1	3	47	73–87	8	9	12	1	2
22	41–48	4	10	10	1	3	48	80–89	16	5	3	3	0
23	45–50	4	11	10	1	3	49	81–90	8	10	10	1	3

5. G. van der Geer, M. van der Vlugt. *Constructing curves over finite fields with many points by solving linear equations*. Preprint 1997.
6. G. van der Geer, M. van der Vlugt. *Tables of curves with many points*. Math. Comp., to appear.
7. J. P. Hansen. *Group codes and algebraic curves*. Mathematica Gottingensis, Schriftenreihe SFB Geometrie und Analysis, Heft 9, 1987.
8. J. P. Hansen, H. Stichtenoth. *Group codes on certain algebraic curves with many rational points*. Appl. Alg. Eng. Commun. Comp. **1** (1990) 67–77.
9. The KANT Group. M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, K. Wildanger. *KANT V4*. J. Symb. Comp. **24/3** (1997) 267–283.
10. K. Lauter. *Deligne Lusztig curves as ray class fields*. Manuscripta Math. **98/1** (1999) 87–96.
11. K. Lauter. *A formula for constructing curves over finite fields with many rational points*. J. Number Theory **74/1** (1999) 56–72.
12. H. Niederreiter, C. P. Xing. *Explicit global function fields over the binary field with many rational places*. Acta Arith. **75/4** (1996) 383–396.
13. H. Niederreiter, C. P. Xing. *Cyclotomic function fields, Hilbert class fields, and global function fields with many rational places*. Acta Arith. **79/1** (1997) 59–76.
14. H. Niederreiter, C. P. Xing. *Algebraic curves over finite fields with many rational points*. Number theory (Eger 1996) 423–443, de Gruyter, Berlin, 1998.
15. H. Niederreiter, C. P. Xing. *Global function fields with many rational places over the ternary field*. Acta Arith. **83/1** (1998) 65–86.
16. H. Niederreiter, C. P. Xing. *A general method of constructing global function fields with many rational places*. Algorithmic Number Theory (Portland 1998), Lect. Notes in Comp. Sci. **1423** 555–566, Springer, Berlin, 1998.
17. M. Perret. *Tours ramifiées infinies de corps de classes*. J. Number Theory **38** (1991) 300–322.
18. M. Rosen. *The Hilbert class field in function fields*. Expo. Math. **5** (1987) 365–378.

19. R. Schoof. *Algebraic curves and coding theory*. UTM **336**, Univ. of Trento, 1990.
20. J.-P. Serre. *Sur le nombre des points rationnelles d'une courbe algébrique sur un corps fini*. C. R. Acad. Sci. Paris **296** (1983) série I, 397–402.
21. J.-P. Serre. *Nombres de points des courbes algébrique sur \mathbb{F}_q* . Séminaire de Théorie des Nombres de Bordeaux **22** (1982/83).
22. J.-P. Serre. *Résumé des cours de 1983–1984*. Annuaire du Collège de France (1984) 79–83.
23. V. Shabat. Unpublished manuscript. University of Amsterdam, 1997/98.
24. H. Stichtenoth. *Algebraic geometric codes associated to Artin-Schreier extensions of $\mathbb{F}_q(z)$* . In: *Proc. 2nd Int. Workshop on Alg. and Comb. Coding Theory*. Leningrad (1990) 203–206.
25. C. P. Xing, H. Niederreiter. *Drinfel'd modules of rank 1 and algebraic curves with many rational points*. Report Austrian Academy of Sciences, Vienna, 1996.

New Results on Lattice Basis Reduction in Practice

Werner Backes¹ and Susanne Wetzel^{*2}

¹ Universität des Saarlandes, Fachbereich 14, Postfach 15 11 50
D-66041 Saarbrücken, Germany
wbackes@cs.uni-sb.de

² Lucent Technologies - Bell Labs, Information Sciences Research
600-700 Mountain Avenue, Murray Hill, NJ 07974, USA
sgwetzel@research.bell-labs.com

Abstract. In this paper we introduce several new heuristics as to speed up known lattice basis reduction methods and improve the quality of the computed reduced lattice basis in practice. We analyze substantial experimental data and to our knowledge, we are the first to present a general heuristic for determining which variant of the reduction algorithm, for varied parameter choices, yields the most efficient reduction strategy for reducing a particular problem instance.

1 Introduction

A lattice is a discrete additive subgroup of \mathbb{R}^n generated by a lattice basis. Lattice basis reduction is the computation of lattice bases consisting of basis vectors which are as small in length as possible. The underlying theory has a long history starting with the reduction of quadratic forms and recently obtained general interest with the introduction of the LLL lattice basis reduction algorithm [13]. It spurred extensive research thus leading to the discovery of important connections of lattice theory with other fields in mathematics and computer science. In particular, the progress in lattice theory has revolutionized combinatorial optimization [9] and cryptography (e.g., [1,2,6,8,11]). Nevertheless, despite the manifold results in theory and the availability of implementations of lattice basis reduction algorithms in various computer algebra systems (e.g., LiDIA [14], Magma [15] or NTL [18]) there is still very little known about the practical performance and strength of lattice basis reduction algorithms. Thus, they are often underestimated as recent results show (e.g., on breaking cryptosystems using lattice basis reduction methods [16,17]). With this paper we close this gap by providing the results of analyzing extensive test data thus supplying useful facts on the practical performance of widely used lattice basis reduction algorithms. Moreover, we introduce newly-developed heuristics designed to improve

^{*} The research was done while the author was a member of the Graduiertenkolleg Informatik at the Universität des Saarlandes, Saarbrücken (Germany), a fellowship program of the DFG (Deutsche Forschungsgemeinschaft).

known lattice basis reduction methods in practice and provide detailed test data on these new methods which are only implemented in LiDIA so far. We restrict the discussion in this paper to the LLL algorithm and its variants as the most well-known and most widely used lattice basis reduction methods in practice.

The outline of the paper is as follows: In Section 2 we give a brief introduction to lattice theory by covering the basic terminology and stating basic auxiliary results in particular of the LLL algorithm, which is the first known polynomial time lattice basis reduction algorithm guaranteed to compute lattice bases consisting of relatively short basis vectors. We then focus on the Schnorr–Euchner algorithm [22] which provided the first essential improvement for making the LLL-reduction algorithm efficiently applicable in practice. In Sections 3 and 4 we introduce (newly-developed) variants of the Schnorr–Euchner reduction algorithm (e.g., based on modular techniques) designed to achieve better run times and reduction results in practice than the classical Schnorr–Euchner algorithm, especially for large lattice bases or bases with large entries. The development of these new heuristics is motivated by the fact that because of the heuristics for preventing and correcting floating point errors, both the run time and the stability of the classical Schnorr–Euchner algorithm strongly depend on the precision of the approximations used. Hence due to stability reasons, for large lattice bases or bases with large entries, a high precision for the approximations has to be used in the classical algorithm thus causing a major loss in efficiency. Section 5 is devoted to the description of suitable test series for testing the different reduction algorithms (i.e., the classical Schnorr–Euchner algorithm as well as the variants presented in Sections 3 and 4) and the analysis of the corresponding substantial experimental data. Among other things, the analysis shows clearly the efficiency of the newly-developed algorithms. Furthermore, based on the data and the analysis, we present a general heuristic for determining which variant of the reduction algorithm, for varied parameter choices, yields the most efficient reduction strategy (with respect to run time or quality of the basis) for reducing a particular problem instance.

2 Background on Lattice Basis Reduction

In this section we present the basic definitions and results which will be used in the sequel. For more details and proofs we refer to [4,9,19,24]. In the following, let $n, k \in \mathbb{N}$ with $k \leq n$. By $\|\underline{b}\|$ we denote the Euclidean length of the column vector \underline{b} and for $z \in \mathbb{R}$, $\lceil z \rceil$ stands for the closest integer to z . An integral lattice $L \subseteq \mathbb{Z}^n$ is defined as $L = \left\{ \sum_{i=1}^k x_i \underline{b}_i \mid x_i \in \mathbb{Z}, i = 1, \dots, k \right\}$, where $\underline{b}_1, \underline{b}_2, \dots, \underline{b}_k \in \mathbb{Z}^n$ are linearly independent vectors. We call $B = (\underline{b}_1, \dots, \underline{b}_k) \in \mathbb{Z}^{n \times k}$ a basis of the lattice $L = L(B)$ with dimension k . Obviously, a lattice has various bases whereas the dimension is uniquely determined. A basis of a lattice is unique up to unimodular transformations such as exchanging two basis vectors, multiplying a basis vector by -1 or adding an integral multiple of one basis vector to another one, for example. The determinant $\det(L) := |\det(B^T B)|^{\frac{1}{2}}$ of the lattice $L \subseteq \mathbb{Z}^n$ with basis $B \in \mathbb{Z}^{n \times k}$ is independent of the choice of the

basis. The Hadamard inequality $\det(L) \leq \prod_{i=1}^k \|b_i\|$ gives an upper bound for the size of the determinant of a lattice. Equality holds iff B is an orthogonal basis. Furthermore, the defect of B is defined as $\text{dft}(B) = \frac{1}{\det(L)} \prod_{i=1}^k \|b_i\|$. In general, $\text{dft}(B) \geq 1$ and $\text{dft}(B) = 1$ iff B is an orthogonal basis. Lattice basis reduction is a technique for reducing (possibly minimizing) the defect of a lattice i.e., it is a technique to construct one of the many bases of a lattice such that the basis vectors are as small as possible (by means of the Euclidean length) and are as orthogonal as possible to each other. The most well-known lattice basis reduction method is the so-called LLL-reduction [13]:

Definition 1. For a lattice $L \subseteq \mathbb{Z}^n$ with basis $B = (\underline{b}_1, \dots, \underline{b}_k) \in \mathbb{Z}^{n \times k}$, corresponding Gram-Schmidt orthogonalization $B^* = (\underline{b}_1^*, \dots, \underline{b}_k^*) \in \mathbb{Q}^{n \times k}$ and Gram-Schmidt coefficients $\mu_{i,j}$ with $1 \leq j < i \leq k$, the basis B is called LLL-reduced if the following conditions are satisfied:

$$|\mu_{i,j}| \leq \frac{1}{2} \quad \text{for } 1 \leq j < i \leq k \quad (1)$$

$$\|\underline{b}_i^* + \mu_{i,i-1}\underline{b}_{i-1}^*\|^2 \geq \frac{3}{4} \|\underline{b}_{i-1}^*\|^2 \quad \text{for } 1 < i \leq k. \quad (2)$$

The first property (1) is the criterion for size-reduction. The constant factor $\frac{3}{4}$ in (2) is the so-called reduction parameter and may be replaced with any fixed real number y with $\frac{1}{4} < y < 1$.

Theorem 1. Let L be a lattice in \mathbb{Z}^n and $B = (\underline{b}_1, \dots, \underline{b}_k) \in \mathbb{Z}^{n \times k}$ be an LLL-reduced basis of L . Then, the following estimates hold:

$$\|\underline{b}_1\| \cdot \dots \cdot \|\underline{b}_k\| \leq 2^{k(k-1)/4} \det(L) \quad (3)$$

$$\|\underline{b}_1\|^2 \leq 2^{k-1} \|\underline{v}\|^2 \quad \text{for all } \underline{v} \in L, \underline{v} \neq \underline{0} \quad (4)$$

For any lattice $L \subseteq \mathbb{Z}^n$ with basis $B = (\underline{b}_1, \dots, \underline{b}_k) \in \mathbb{Z}^{n \times k}$, the LLL-reduction of the basis B can be computed in polynomial time. More precisely, the number of arithmetic operations needed by the LLL algorithm is $O(k^3 n \log C)$, and the integers on which these operations are performed each have binary size $O(k \log C)$ where $C \in \mathbb{R}$, $C \geq 2$ with $\|\underline{b}_i\|^2 \leq C$ for $1 \leq i \leq k$. Thus, from a theoretical point of view, the algorithm performs very well since it yields a reasonably good reduction result within polynomial time. In practice however, the classical algorithm [13] suffers from the slowness of the subroutines for the exact long integer arithmetic which has to be used in order to guarantee that no errors occur in the basis (thus changing the lattice). Speeding up the algorithm by simply doing the operations in floating point arithmetic results in an unstable algorithm due to occurring floating point errors and error propagation. In [22], Schnorr and Euchner have rewritten the original LLL algorithm in such a way that an approximation of the integer lattice with a faster floating point arithmetic is only used for the computation of the Gram-Schmidt coefficients $\mu_{i,j}$ ($1 \leq j < i \leq k$) while all the other operations are done on the integer lattice using an exact

integer arithmetic. Moreover, Schnorr and Euchner have introduced heuristics for avoiding and correcting floating point errors, thus inventing a practical floating point variant of the original algorithm with good stability, which allows a tremendous speed-up in the computation of an LLL-reduced lattice basis. Nevertheless, for large lattice bases or bases with large entries, the algorithm still lacks of efficiency due to the fact that high precision approximations have to be used for stability reasons.

Therefore, before focusing on the presentation and discussion of practical results achieved by using the Schnorr–Euchner reduction algorithm in different settings (Section 5), thus comparing theory and practical performance of this lattice basis reduction algorithm, we will in the following (Sections 3 and 4) first present (new) heuristics designed to further speed up the reduction (such that even larger lattice bases with bigger entries can be reduced in a reasonable amount of time) and improve the quality of the reduction results (i.e., computing reduced lattice bases consisting of shorter lattice vectors in comparison with the reduction results obtained by the classical Schnorr–Euchner algorithm). Due to the page limit we can only sketch the basic ideas and refer to [24] for a detailed description.

3 Heuristics to Achieve an Additional Speed-Up

In this section we will introduce heuristics designed to allow a speed-up of the computation in comparison to the classical Schnorr–Euchner algorithm [22]. The first heuristic in this setting, the so-called late size-reduction heuristic, is motivated by the following observation: While at stage l ($2 \leq l \leq k$) of the LLL-reduction process the Gram-Schmidt coefficients $\mu_{l,m}$ ($1 \leq m \leq l-2$) have to be size-reduced only if the basis vectors b_l and b_{l-1} are not swapped [13], in the classical Schnorr–Euchner algorithm always all the Gram-Schmidt coefficients $\mu_{l,m}$ ($1 \leq m \leq l-1$) are size-reduced. Therefore, a heuristic to speed up the LLL-reduction process in practice can be stated as follows:

Heuristic 1 (Late Size-Reduction). *Before checking the LLL condition (2) at stage l of the reduction process [22], size-reduce only the Gram-Schmidt coefficient $\mu_{l,l-1}$. Perform the size-reduction of the other coefficients $\mu_{l,m}$ with $1 \leq m \leq l-2$ only if neither the stage index has to be decreased (due to accumulated floating point errors) nor the basis vectors b_l and b_{l-1} have to swapped.*

While the late size-reduction heuristic centers on the time when the size-reductions are performed, another new heuristic, called modified size-reduction heuristic, focuses on the way the size-reductions are done in order to speed up the reduction of a lattice basis:

Heuristic 2 (Modified Size-Reduction). *At the beginning of each size-reduction step (see [22]) check whether $|\lceil \mu_{i,j} \rceil|$ ($1 \leq j < i \leq k$) is larger than a certain bound. If so, perform a correction step and simplify the actual size-reduction by approximating $|\lceil \mu_{i,j} \rceil|$ with 2^t where $t = \lceil \log(|\lceil \mu_{i,j} \rceil|) \rceil$, thus replacing the original size-reduction with a fast shift operation.*

Whereas the heuristics introduced so far simplify or eliminate unnecessary operations, a different approach to reduce the overall run time can be taken by doing the computations on shorter operands thus allowing approximations with a lower precision than in the classical Schnorr–Euchner algorithm. The first heuristic in this category employs an iterative technique similar to the ones used in [12] and [21] for speeding up Euclid’s algorithm and the reduction of quadratic forms, respectively:

Heuristic 3 (Iterative Heuristic). *For reducing a given lattice basis, first work only with the leading digits of each entry of the basis vectors. Then, apply the performed reduction steps also to the original lattice basis and compute the final LLL-reduced lattice basis.*

The following theorem is a generalization of the idea presented in [20,23] to arbitrarily chosen lattice bases. It shows in detail how the described heuristic is formalized and how it can be iterated:

Theorem 2. *Let $B = (b_{i,j}) \in \mathbb{Z}^{n \times k}$ be a basis of lattice L and $u, v \in \mathbb{N}$ be such that $\lfloor \log_2(b_{i,j}) \rfloor + 1 \leq u \cdot v$ ($1 \leq i \leq n$, $1 \leq j \leq k$). Moreover, for $1 \leq i \leq n$, $1 \leq j \leq k$ and $1 \leq t \leq u$ let $b_{i,j}^{(t)} = \left\lfloor \frac{b_{i,j}}{2^{v(u-t)}} \right\rfloor$ and $B^{(t)} = (b_{i,j}^{(t)})$, let $D^{(t)} = (d_{i,j}^{(t)})$ be defined by $b_{i,j}^{(t+1)} = 2^v b_{i,j}^{(t)} + d_{i,j}^{(t)}$ and let $E = 2^v I_k$. With $T^{(0)} = I_k$, $C^{(1)} = B^{(1)}T^{(0)} = B^{(1)}$, $R^{(t)} = \text{LLL}(C^{(t)}) = C^{(t)}T_t$, $T^{(t)} = T^{(t-1)}T_t$ as well as $C^{(t+1)} = R^{(t)}E + D^{(t)}T^{(t)}$, then $R^{(u)} = \text{LLL}(B) = BT^{(u)}$. (T_t is the transformation matrix computed in the course of the reduction process [13,22].)*

Thus, for reducing the basis B , the iterative reduction algorithm applies the classical Schnorr–Euchner algorithm u times to the generating systems $C^{(1)}, \dots, C^{(u)}$. In each iteration step, v additional digits of the original input data are included in the computation. Hence, the result of the last iteration step yields the LLL-reduction of the lattice basis B .

The last heuristic presented in this section is based on modular computations. From other areas in algorithmic number theory we know that the application of modular techniques has been instrumental in bringing about much more efficient solutions to well-known problems such as the computation of the determinant [4] or the Hermite normal form [7] of integer matrices. This is due to the fact that with modular techniques most of the computations can be performed on operands which are much smaller than the ones occurring in the conventional algorithms. The following considerations show how modular techniques can also be applied to the problem of computing an LLL-reduced lattice basis, thus allowing an improvement of the run time of the classical reduction algorithm:

Theorem 3 ([7]). *Let $L \subseteq \mathbb{Z}^n$ be an n -dimensional lattice and $\Delta = z \cdot \det(L)$ with $z \in \mathbb{Z}$ be a multiple of the lattice determinant. Then, $\Delta \underline{e}_i \in L$ for $1 \leq i \leq n$.*

Lemma 1. *Let $L \subseteq \mathbb{Z}^n$ be an n -dimensional lattice with basis $B = (b_1, \dots, b_n) \in \mathbb{Z}^{n \times n}$ and $\Delta = z \cdot \det(L)$ where $z \in \mathbb{Z}$. Then, $L(b_1, \dots, b_n) = L(\underline{b}_1 \bmod \Delta, \dots, \underline{b}_n \bmod \Delta, \Delta \underline{e}_1, \dots, \Delta \underline{e}_n)$.*

This leads to the following heuristic, which can be implemented in various ways [24]:

Heuristic 4. *At first reduce the basis of the n -dimensional lattice $L \subseteq \mathbb{Z}^n$ modulo Δ , a multiple of the determinant, thus obtaining a system of generating vectors $(\underline{b}_1 \bmod \Delta, \dots, \underline{b}_n \bmod \Delta)$. Apply to that system of generating vectors a modular variant of the Schnorr–Euchner algorithm where additional modular operations are performed during the size-reduction process. Compute the LLL-reduced basis of the lattice $L(\underline{b}_1, \dots, \underline{b}_n)$ by applying the Schnorr–Euchner algorithm to the system of generating vectors consisting of the resulting vectors of the reduction with the modular Schnorr–Euchner algorithm as well as the vectors $\Delta e_1, \dots, \Delta e_n$.*

4 Heuristics to Achieve Better Reduction Results

After introducing heuristics for speeding up the computation of a reduced lattice basis, we will now present two heuristics where the improvement of the quality of the reduction result is the main objective in their development, possibly even at the expense of the run time. The first heuristic, the so-called deep insertion heuristic, is due to Schnorr and Euchner [22]:

Heuristic 5 (Deep Insertion). *For checking the LLL-reduction condition (2) at stage l take into consideration not only the values $\|\underline{b}_{l-1}^*\|$ and $\|\underline{b}_l^*\|$ (as it was done in the LLL algorithm and the classical Schnorr–Euchner algorithm) but also the earlier $\|b_j^*\|$'s with $1 \leq j \leq l-2$ by extending the exchange of b_l and b_{l-1} to a deep insertion step, i.e., inserting b_l at the best possible position i within the index interval $[1, \dots, l-1]$.*

Applying this heuristic, short orthogonal vectors are found further left in the orthogonalization thus yielding shorter basis vectors in the reduced basis than in the case of using the classical Schnorr–Euchner algorithm. It has to be noted that only a certain amount of deep insertion steps can be performed in order to guarantee polynomial run time of the reduction process (for details see [22]).

A new heuristic to achieve a better reduction result is motivated by the following example:

Example 1. Let $B = \begin{pmatrix} 10 & -9 & 18 \\ 0 & 10 & 45 \\ 10 & 11 & -12 \end{pmatrix}$ be a basis of a lattice $L \subseteq \mathbb{Z}^3$. The basis

B is already LLL-reduced with reduction parameter $y \in (\frac{1}{4}, 1)$, $\|\underline{b}_1\|^2 = 200$, $\|\underline{b}_2\|^2 = 302$ and $\|\underline{b}_3\|^2 = 2493$. However, performing an additional size-reduction step even though $|\mu_{3,2}| = 0.5$ (i.e., size-reduction does not change $|\mu_{3,2}|$) results in a shorter basis vector $\underline{b}_3 = (27, 35, -23)^T$ with $\|\underline{b}_3\|^2 = 2483$ and thus in a better LLL-reduced lattice basis.

Heuristic 6 (Special Case for μ). *If $|\mu_{i,j}| = 0.5$ with $1 \leq j < i \leq k$ perform a size-reduction step iff $\|\underline{b}_i - \text{sign}(\mu_{i,j})\underline{b}_j\| < \|\underline{b}_i\|$.*

5 Tests and General Heuristic

Based on comprehensive tests, we will now analyze the practical performance of the classical Schnorr–Euchner algorithm as well as the algorithms implementing the (new) heuristics thus providing essential new insight into the practical performance of lattice basis reduction algorithm aside from the known theoretical results. Moreover, we present a general heuristics for the use of these lattice basis reduction algorithms.

For the tests we have used three generic test classes, namely knapsack lattices, unimodular lattices, and random lattices. Knapsack lattices arise in the context of solving knapsack problems [6] and are of the form $L(B) \subseteq \mathbb{Z}^{(n+3)}$ where

$$B = (\underline{b}_1, \dots, \underline{b}_{n+1}) = \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 2 & 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 2 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 2 & 1 \\ a_1 W & a_2 W & \cdots & a_{n-1} W & a_n W & SW \\ 0 & 0 & \cdots & 0 & 0 & -1 \\ W & W & \cdots & W & W & \frac{n}{2} W \end{pmatrix} \quad (5)$$

with positive integer weights a_i ($1 \leq i \leq n$), a sum $S \in \mathbb{N}$ and $W > \sqrt{n}$. This test class has been chosen due to the fact that lattice bases in various contexts (e.g., finding a small root of univariate modular equations [5], gcd computations [10], factoring, Diophantine equations [13]) have the same structure as the bases of knapsack lattices. The unimodular lattices are generated by a unimodular basis B , i.e., $B \in \mathbb{Z}^{(n \times n)}$ with $|\det(B)| = 1$, the random $(n \times n)$ -lattices $L \subseteq \mathbb{Z}^n$ are generated by a randomly chosen basis $B \in \mathbb{Z}^{(n \times n)}$. In the sequel, for simplicity we shall use the notion “knapsack lattice bases” etc. instead of “bases of knapsack lattices” etc. even though we are aware of the fact that it is not absolutely correct in a mathematical sense.

All tests have been done using the implementations of lattice basis reduction methods available in the computer algebra LiDIA [3,14]. While there are various implementations of the classical Schnorr–Euchner algorithm (e.g., in computer algebra systems such as LiDIA, Magma [15] and NTL [18]), so far implementations of the heuristics presented in Sections 3 and 4 are only available in LiDIA.

The tests have been performed on Sparc 4’s with 110 MHz and 32 MB main memory. For a detailed description of the test instances and the choice of the test parameters we refer to [24].

5.1 Tests of the Different Variants

5.1.1 Performance and Quality. At first we focus on the general performance of the classical Schnorr–Euchner algorithm, the variant of doing deep insertions, the algorithm of applying late size-reduction, the algorithm using the modified size-reduction and the variant considering the special case $|\mu_{i,j}| = 0.5$. The computations were done with reduction parameter $y = 0.99$ using doubles

for the approximations [22]. The tests have been performed for different choices of n (e.g., $n = 10, \dots, 150$) and various sizes of the entries (e.g., bit length $b = 20, \dots, 300$).

The test results [24] show that for a fixed dimension n the run time of the algorithms increases as the density of the knapsack decreases. For a fixed density the run time for reducing knapsack lattice bases increases as the dimension increases. These characteristics are due to the fact that the run time of the algorithms depends both on the dimension of the lattice to be reduced and the size of the input data such that it increases as the dimension or the size of the entries of the lattice basis vectors grow. Furthermore, the reduction of knapsack lattice bases always results in a vast decrease of the average length of the basis vectors and the defect.

For randomly chosen lattice bases the reduction time is relatively small (in comparison with the other test instances) and depends mainly on the dimension of the lattice. This is due to the fact that any randomly chosen lattice basis is already significantly reduced, thus the reduction will result only in a small decrease of the average length of the basis vectors. This can be explained by the observation that due to the Gaussian heuristic, the expected length of a smallest vector in a random lattice of dimension n with determinant Δ lies between $\Delta^{1/n} \sqrt{\frac{n}{2\pi e}}$ and $\Delta^{1/n} \sqrt{\frac{n}{\pi e}}$. In the case of reducing unimodular lattice bases, the reduction time mainly depends on the dimension of the lattice.

Checking whether an additional size-reduction step should be performed if $|\mu_{i,j}| = 0.5$ causes an increase of the run time but in general does not yield a shorter average length or a smaller defect. However, in the case of reducing knapsack lattice bases, the increase is negligible. In general, for random lattice bases the special case $|\mu_{i,j}| = 0.5$ does not occur. This is due to the fact that only few reduction steps are performed anyway (in comparison with the other test instances) and therefore it is very unlikely that $|\mu_{i,j}| = 0.5$ occurs.

Except for random lattice bases the deep insertion mechanism causes a major increase in the run time but also results in a great improvement of the average length of the vectors of the reduced bases as well as a better defect for knapsack lattice bases. In Figure 1, the influence of the deep insertion mechanism is illustrated on example of the first five basis vectors of reduced knapsack lattice bases with $n = 150$. In the case of random lattice bases, only few deep insertion steps are performed since these lattice bases are already quite well reduced from the beginning. Thus, there is hardly any decrease of the defect or the average length in comparison with the results of the classical algorithm. Logically, for knapsack and unimodular lattice bases, the amount of deep insertions increases with the dimension of the lattice. This is due to the also increasing amount of reduction steps which simply implies a higher probability that deep insertions can be performed.

As for the late size-reduction variant it turns out that this algorithm performs well for small-dimensional test lattices ($n < 30$) but lacks stability otherwise, even though provisions for correcting floating point errors and preventing error propagation were already taken. For stability reasons it seems to be crucial

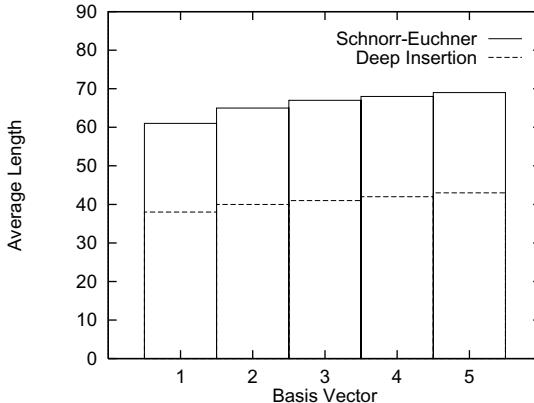


Fig. 1. Different Variants: Deep Insertion Heuristic

that at stage l of the reduction process all Gram-Schmidt coefficients $\mu_{l,j}$ with $1 \leq j \leq l-1$ are size-reduced before a possible step back might occur (due to a large size-reduction coefficient), thus allowing a faster decrease in the size of the intermediate results as it is the case in the late size-reduction algorithm.

Applying the modified size-reduction algorithm for reducing the lattice bases causes no stability problems. But even though the size-reduction process was simplified, i.e., in the case of a large reduction coefficient, the original size-reductions were replaced with simple shift operations, it turned out that in most cases the heuristic does not improve the reduction time. This is due to the fact that in many cases, the shift operations are not accurate enough in a sense that after size-reducing the Gram-Schmidt coefficient $\mu_{i,j}$, $|\mu_{i,j}|$ is still far off from being less or equal to 0.5, thus causing additional operations.

In summary, one may say that for reducing unimodular lattice bases neither the application of the deep insertion mechanism nor the checks on the special case $|\mu_{i,j}| = 0.5$ are useful since already the classical Schnorr–Euchner algorithm yields a minimal basis for unimodular lattices and none of the mechanisms yields an advantage with respect to the run time. For knapsack lattice bases the additional checks in the case of $|\mu_{i,j}| = 0.5$ make no big difference in the run time but might yield shorter basis vectors in some cases, thus supporting the application of this heuristic in practice. Using the deep insertion mechanism for reducing knapsack lattice bases has the disadvantage of increasing the run time but also the advantage of a decrease of the defect and average length of the basis vectors.

5.1.2 Limits. In the following, we will concentrate on tests of the limits of the classical Schnorr–Euchner algorithm and how improvements can be achieved by using the newly-developed modular and the iterative variant of the classical reduction algorithm. In this context, limits are either meant to be the bounds at which the other variants begin to out-perform the classical algorithm or the bounds from which on `xdoubles` (floating point arithmetic with twice the preci-

sion of **doubles**) or even **bigfloats** (multi-precision floating point arithmetic, see also [14]) have to be used for doing the approximations in the classical Schnorr–Euchner algorithm in order to guarantee that the algorithm will terminate and yield an LLL-reduced lattice basis while in the case of the modular and iterative variants **doubles** are still sufficient for achieving the same result. It is important to know these limits since both the size of the input data as well as the approximations used affect the run time of the reduction algorithm fundamentally.

In the first test scenario, we have applied the classical Schnorr–Euchner algorithm doing the approximations with **doubles**, **xdoubles** and **bigfloats** and the iterative algorithm with $v = 0.25b$, $v = 0.33b$, $v = 0.5b$ and $v = 0.75b$ (doing the approximations with **doubles**) to knapsack lattice bases with $n = 100, 110, \dots, 200$ and bit lengths $b = n, 2n, 4n$. The reduction parameter was chosen as $y = 0.99$.

In the second set-up, we were focusing on reducing lattice bases $B \in \mathbb{Z}^{n \times n}$ with large entries where the corresponding lattice $L(B)$ has a small determinant $\det(L) = \Delta$, i.e., $n = 10, \dots, 100$, $b = 200, 400$ and $\Delta \in [1, 2^{32}]$. The tests were performed using the classical Schnorr–Euchner algorithm, the modular variant Modular_1 and Modular_2 = Schnorr–Euchner($\underline{b}_1 \bmod \Delta, \dots, \underline{b}_n \bmod \Delta, \Delta I_n$) (doing the approximations by means of **doubles** and using reduction parameter $y = 0.99$).

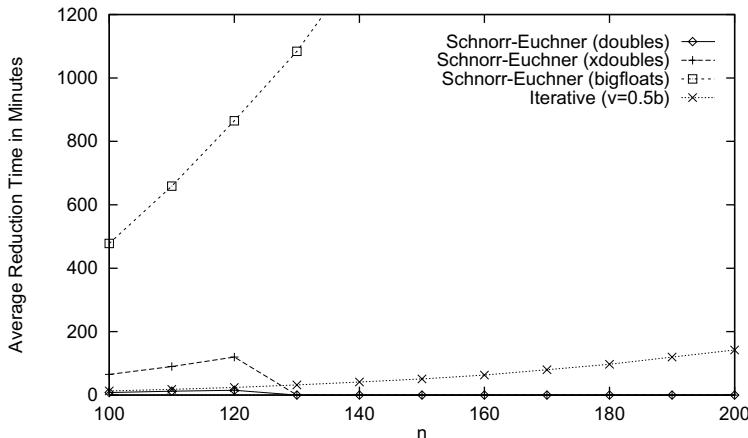


Fig. 2. Different Variants: Knapsack Lattice Bases ($b = 4n$)

The tests on knapsack lattice bases show (see Appendix and [24]) that using the Schnorr–Euchner reduction algorithm and doing the approximations by means of **doubles** works well for lattice bases with $b = n$ and $b = 2n$. In the case of $b = 4n$ and starting at $n = 130$, the approximations using **doubles** are no longer exact enough, thus resulting in a non-reduced basis. At the same time,

using `xdoubles` is not sufficient either. This is due to the fact that `xdoubles` have twice the precision of `doubles` but no larger exponent [14]. Consequently, not only the precision but also the size of the exponent of the approximation is crucial for the stability of the algorithm. Using `bigfloats` with four times the precision of `doubles` and an enlarged exponent increases the reduction time considerably but yields a correctly reduced lattice basis. A major improvement can be achieved by using the iterative algorithms (see Figure 2).

The results of the iterative variant, choosing $v = 0.25b$, $v = 0.33b$ and $v = 0.5b$ (v is the amount of additional digits of the original input data which are included in each new iteration step) show that doing the approximations with `doubles` is sufficient for all test classes. However, since several lattice basis reductions have to be done in the course of the iterative algorithm, the iterative algorithm does not out-perform the classical Schnorr–Euchner algorithm until the Schnorr–Euchner algorithm requires `xdoubles` (`bigfloats`) for the approximations while the iterative variant still works with `doubles` (`xdoubles`). These behavioral characteristics of the iterative variant and the classical Schnorr–Euchner algorithm (in combination with `doubles`, `xdoubles` and `bigfloats` for the approximations) are illustrated in Figure 2 for knapsack lattice bases with $b = 4n$.

In Figure 2 it can be seen that as long as the approximations are done with `doubles` the run time of the iterative implementation is about twice that of the Schnorr–Euchner algorithm. Furthermore, the data for the iterative lattice basis

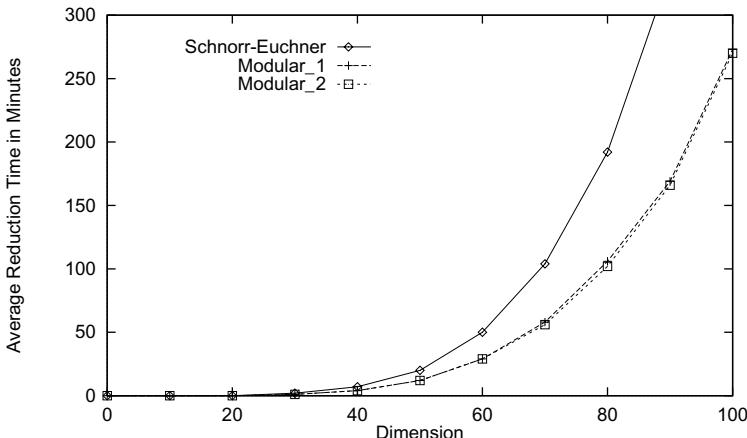


Fig. 3. Different Variants: Lattices with Small Determinant

reduction algorithms show that the reduction time decreases as v increases since a large v requires fewer iterations. At the same time, the computations have to be performed on larger operands, thus possibly causing the same problems as in the case of using the classical Schnorr–Euchner algorithm. For example, for $v = 0.75$, $n \geq 170$ and $b = 4n$ it is no longer sufficient to do the approximations

by means of **doubles**. On the other hand, for large lattice bases with huge entries, the run time decreases as v decreases. In this case, the advantage that the computations can be done on small operands predominates the disadvantage that many iterations have to be performed. Hence, these observations show that the size of v and thus the amount of iterations has to be chosen skillfully in order to obtain the best possible results.

The modular variants out-perform the classical Schnorr–Euchner algorithm for increasing b . This is due to the fact that using the modular variants, the operands on which the computations are performed are much smaller in size than in the case of applying the classical Schnorr–Euchner algorithm. For increasing b this advantage compensates the disadvantage of the additional computations in the case of the modular variants, necessary in order to guarantee that the correct LLL-reduced basis is computed in any case. The performance of the algorithms is impressively demonstrated in Figure 3 for lattices with a small determinant and $b = 400$. Figure 3 also shows that the differences in the run times of the modular variants are small. Consequently, after the initial modular reduction of the lattice only few additional modular reductions have to be performed in the course of the size-reductions in the algorithm **Modular_1**.

To sum up, one may say that for large dimensions or large entries of the lattice bases, the newly-proposed modular and the iterative variants out-perform the classical Schnorr–Euchner algorithm.

5.2 Additional Tests

In addition to the tests described so far, experiments with varied reduction parameters respectively series of reduction parameters and varied scalar products have been performed. Moreover, the use of performing the reductions based on the Gram matrix instead of the original basis of the lattice was tested [24].

In summary, it may be said that for achieving the best reduction results in the case of knapsack lattice bases, it is recommended to apply a sequence of reductions with varied reduction parameters. For reducing unimodular and random lattice bases in general it is sufficient to use the reduction parameter $y = 0.75$. By using a weighted scalar product to reduce a given lattice basis it is possible to decrease the row sum norm of a particular row of that lattice basis significantly. Moreover, the Gram heuristic is only advisable for reducing unimodular lattice bases.

5.3 General Heuristic

Based on the comprehensive analysis of the test results we can now deduce heuristics for the general use of the different variants of the Schnorr–Euchner algorithm in order to achieve the best possible reduction results or to minimize the necessary reduction time:

In the case of reducing knapsack lattice bases, it has to be noted that in general these two goals cannot be achieved at the same time. Generally, the following heuristic is suggested:

Heuristic 7. *In order to minimize the run time for reducing small knapsack lattice bases use the classical Schnorr–Euchner algorithm in combination with a sequence of reduction parameters doing the approximations with doubles. For large lattice bases or lattice bases where the size of the entries is large (more than 400 bits) use the iterative algorithm also in combination with a sequence of reduction parameters. In order to maximize the quality of the reduction of bases corresponding to knapsack lattices apply the deep insertion heuristic doing the approximations with doubles. For large lattice bases with huge entries (more than 400 bits) use the iterative algorithm in combination with the deep insertion mechanism.*

Since the tests have shown that randomly chosen lattice bases are already significantly reduced, it is suggested to LLL-reduce them in the following way:

Heuristic 8. *For reducing randomly chosen lattice bases use classical Schnorr–Euchner algorithm with $y = 0.75$ doing the approximations by means of doubles. If the determinant is known for large $(n \times n)$ -lattice bases, or $(n \times n)$ -lattice bases with large entries, apply the modular reduction variant. Otherwise, use the classical Schnorr–Euchner algorithm and adjust the approximations if necessary.*

In the case of unimodular lattice bases, the following heuristic should be applied:

Heuristic 9. *For reducing unimodular lattice bases compute the corresponding Gram matrix and apply the Schnorr–Euchner reduction algorithm with reduction parameter $y = 0.75$ and doubles for the approximations. For large lattice bases with huge entries adjust the approximations if necessary.*

For a given lattice basis which does not necessarily belong to any of the classes discussed so far we suggest the following method for reducing the given basis:

Heuristic 10. *If the given lattice basis is sparse, apply the proposed heuristics for knapsack lattice bases. In the case of a dense lattice basis, proceed as in the case of randomly chosen lattices.*

If the run time is not of importance, the quality of the reduction of a lattice basis can be even further improved by repeatedly sorting and mixing up the reduced basis (by performing weight-reductions, permuting the basis randomly or using so-called Hadamard transformation matrices) and reducing the basis again [24].

6 Conclusions

In this paper, we have presented various new heuristics and analyzed a comprehensive series of tests on the practical performance of the different variants of the Schnorr–Euchner algorithm. Based on these results, we have introduced heuristics for the general application of these lattice basis reduction algorithms designed to minimize the reduction time or achieve a best possible reduction result in practice. For any given reduction algorithm (e.g., Korkine-Zolotarev-reduction [19]), we believe that in order to draw similar conclusions about the best reduction strategy, experimentation and analysis similar to that presented herein must be performed.

References

1. Ajtai, M.: *Generating Hard Instances of Lattice Problems*. Proceedings of the 28th ACM Symposium on Theory of Computing, pp. 99–108 (1996).
2. Ajtai, M., and Dwork, C.: *A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence*. Proceedings of the 29th ACM Symposium on Theory of Computing, pp. 284–293 (1997).
3. Biehl, I., Buchmann, J., and Papanikolaou, T.: *LiDIA: A Library for Computational Number Theory*. Technical Report 03/95, SFB 124, Universität des Saarlandes, Saarbrücken, Germany (1995).
4. Cohen, H.: *A Course in Computational Algebraic Number Theory*. Second Edition, Springer-Verlag, Heidelberg (1993).
5. Coppersmith, D.: *Finding a Small Root of a Univariate Modular Equation*. Proceedings of EUROCRYPT '96, Springer Lecture Notes in Computer Science LNCS 1070, pp. 155–165 (1996).
6. Coster, M.J., Joux, A., LaMacchia, B.A., Odlyzko, A.M., Schnorr, C.P., and Stern, J.: *Improved Low-Density Subset Sum Algorithms*. Journal of Computational Complexity, Vol. 2, pp. 111–128 (1992).
7. Domich, P.D., Kannan, R., and Trotter, L.E.: *Hermite Normal Form Computation using Modulo Determinant Arithmetic*. Mathematics Operations Research, Vol. 12, No. 1, pp. 50–59 (1987).
8. Goldreich, O., Goldwasser, S., and Halevi, S.: *Public-Key-Cryptosystems from Lattice Reduction Problems*. Proceedings of CRYPTO '97, Springer Lecture Notes in Computer Science LNCS 1294, pp. 112–131 (1997).
9. Grötschel, M., Lovász, L., and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*. Second Edition, Springer-Verlag, Heidelberg (1993).
10. Havas, G., Majewski, B.S., and Matthews, K.R.: *Extended GCD Algorithms*. Technical Report TR0302, The University of Queensland, Brisbane, Australia (1994).
11. Joux, A., and Stern, J.: *Lattice Reduction: A Toolbox for the Cryptanalyst*. Journal of Cryptology, Vol. 11, No. 3, pp. 161–185 (1998).
12. Knuth, D.E.: *The Art of Computer Programming. Volume 2: Seminumerical algorithms*. Second Edition, Addison-Wesley, Reading, Massachusetts (1981).
13. Lenstra, A.K., Lenstra, H.W., and Lovász, L.: *Factoring Polynomials with Rational Coefficients*. Math. Ann. 261, pp. 515–534 (1982).
14. LiDIA Group: *LiDIA Manual*. Universität des Saarlandes/TU Darmstadt, Germany, see LiDIA homepage: <http://www.informatik.tu-darmstadt.de/TI/LiDIA> (1999).
15. Magma homepage (1999):
<http://www.maths.usyd.edu.au:8000/comp/magma/Overview.html>.
16. Nguyen, P.: *Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97*. Proceedings of Crypto '99, Springer Lecture Notes in Computer Science LNCS 1666, pp. 288–304 (1999).
17. Nguyen, P., and Stern, J.: *Cryptanalysis of a Fast Public Key Cryptosystem Presented at SAC '97*. Proceedings of Selected Areas in Cryptography '98, Springer Lecture Notes in Computer Science LNCS 1556 (1999).
18. NTL homepage: <http://www.cs.wisc.edu/~shoup/ntl> (1999).
19. Pohst, M.E., and Zassenhaus, H.J.: *Algorithmic Algebraic Number Theory*. Cambridge University Press (1989).
20. Radziszowski, S., and Kreher, D.L.: *Solving Subset Sum Problems with the L^3 Algorithm*. J. Combin. Math. Combin. Computation, Vol. 3, pp. 49–63 (1988).

21. Rickert, N.W.: *Efficient Reduction of Quadratic Forms*. Proceedings of Computers and Mathematics '89, pp. 135–139 (1989).
22. Schnorr, C.P., and Euchner, M.: *Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems*. Proceedings of Fundamentals of Computation Theory '91, Springer Lecture Notes in Computer Science LNCS 529, pp. 68–85 (1991).
23. de Weger, B.: *Algorithms for Diophantine Equations*. PhD Thesis, Centrum voor Wiskunde en Informatica, Amsterdam, Netherlands (1988).
24. Wetzel, S.: Lattice Basis Reduction Algorithms and their Applications. PhD Thesis, Universität des Saarlandes, Saarbrücken, Germany (1998).

Appendix: Tests

In the following, we provide some of the test data [24] for the tests described and analyzed in Section 5. The general notation used is as follows:

determinant	determinant of the lattice
defect	defect of the lattice basis
av. length	average length of the base vectors
factor	$\text{factor} = (\text{density})^{-1}$
time in (m)s	time in (milli)seconds needed for reducing the lattice
reduction steps	number of size-reductions performed
correction steps	number of approximations performed
swaps	number of exchanges
step backs	number of occurring decreases of the stage index
exact SP	number of exactly computed scalar products
LLL	original Schnorr–Euchner algorithm - doing the approximations with <code>doubles</code> ($y = 0.99$ unless otherwise stated)
LLL_2	original Schnorr–Euchner algorithm - doing the approximations with <code>xdoubles</code> ($y = 0.99$)
LLL_4	original Schnorr–Euchner algorithm - doing the approximations with <code>bigfloats</code> having four times the precision of <code>doubles</code> as well as an enlarged exponent ($y = 0.99$)
LLL_Iterative_0.50	iterative variation of the Schnorr–Euchner algorithm with $v = 0.50b$ and doing the approximations with <code>doubles</code> ($y = 0.99$)
Modular_1	modular variation of the Schnorr–Euchner algorithm doing the approximations with <code>doubles</code> ($y = 0.99$)
Modular_2	= Schnorr–Euchner_Generate($\underline{b}_1 \bmod \Delta, \dots, \underline{b}_n \bmod \Delta, \Delta I_n$) with $y = 0.99$, $\Delta = \det(L(B))$ and doing the approximations with <code>doubles</code>
Knapsack	knapsack lattices
Random_Det	lattices generated using a modification of the LiDIA function <code>randomize_with_det</code>

Note that the figures provided are rounded values corresponding to the original results.

A.1 Input Data and Test Results

Knapsack, av. length												
dim	factor ≈ 1.0				factor ≈ 2.0				factor ≈ 4.0			
	0	max	min									
100	1 · 10 ⁻³⁰	1 · 10 ⁻³¹	8 · 10 ⁻³⁰	1 · 10 ⁻³¹	2 · 10 ⁻³¹	1 · 10 ⁻³¹	1 · 10 ⁻³¹					
110	1 · 10 ⁻³⁴	1 · 10 ⁻³⁴	9 · 10 ⁻³³	1 · 10 ⁻³⁴	2 · 10 ⁻³³	2 · 10 ⁻³³	2 · 10 ⁻³³					
120	1 · 10 ⁻³⁷	2 · 10 ⁻³⁵	3 · 10 ⁻³⁵	2 · 10 ⁻³⁵	2 · 10 ⁻³⁵							
130	1 · 10 ⁻⁴⁰	3 · 10 ⁻³⁷	3 · 10 ⁻³⁷	2 · 10 ⁻³⁷	2 · 10 ⁻³⁷							
140	1 · 10 ⁻⁴³	1 · 10 ⁻³⁵	3 · 10 ⁻³⁶	3 · 10 ⁻³⁶	2 · 10 ⁻³⁶							
150	1 · 10 ⁻⁴⁶	2 · 10 ⁻⁴⁹	1 · 10 ⁻³⁹									
160	1 · 10 ⁻⁴⁹	2 · 10 ⁻⁵²	2 · 10 ⁻⁴³	4 · 10 ⁻⁴³	4 · 10 ⁻⁴³	4 · 10 ⁻⁴³						
170	1 · 10 ⁻⁵²	2 · 10 ⁻⁵⁵	2 · 10 ⁻⁴⁵	5 · 10 ⁻⁴⁵	5 · 10 ⁻⁴⁵	5 · 10 ⁻⁴⁵						
180	1 · 10 ⁻⁵⁵	2 · 10 ⁻⁵⁸	2 · 10 ⁻⁴⁸	6 · 10 ⁻⁴⁸	6 · 10 ⁻⁴⁸	6 · 10 ⁻⁴⁸						
190	1 · 10 ⁻⁵⁸	2 · 10 ⁻⁶¹	2 · 10 ⁻⁴⁹	6 · 10 ⁻⁴⁹	6 · 10 ⁻⁴⁹	6 · 10 ⁻⁴⁹						
200	1 · 10 ⁻⁶¹	2 · 10 ⁻⁶⁴	7 · 10 ⁻⁵¹									

Random, Lüt: 400 bits												
dim	0				determinant				defect			
	0	max	min	0	max	min	0	max	min	0	max	min
010	3 · 10 ⁻¹²⁰	4 · 10 ⁻¹²⁰	2 · 10 ⁻¹²⁰	7 · 10 ⁻¹²⁰	43 0008	63 616	55 550	4 · 10 ⁻¹¹⁴	3 · 10 ⁻¹¹⁴	1 · 10 ⁻¹¹⁴	1 · 10 ⁻¹¹⁴	1 · 10 ⁻¹¹⁴
020	8 · 10 ⁻¹²⁰	9 · 10 ⁻¹²⁰	1 · 10 ⁻¹²⁰	50 0537	64 890	20 0001	1 · 10 ⁻²³⁵⁴	9 · 10 ⁻²³⁵⁴	9 · 10 ⁻²³⁵⁴	9 · 10 ⁻²³⁵⁴	9 · 10 ⁻²³⁵⁴	9 · 10 ⁻²³⁵⁴
030	1 · 10 ⁻¹²¹	44 5589	65 016	22 8487	4 · 10 ⁻³⁵⁶⁸	2 · 10 ⁻³⁵⁶⁹	9 · 10 ⁻³⁵⁶⁶	9 · 10 ⁻³⁵⁶⁶	9 · 10 ⁻³⁵⁶⁶			
040	1 · 10 ⁻¹²¹	35 726	64 248	43 35	1 · 10 ⁻⁴⁷⁸⁵	5 · 10 ⁻⁴⁷⁸⁵	8 · 10 ⁻⁴⁷⁸²	8 · 10 ⁻⁴⁷⁸²	8 · 10 ⁻⁴⁷⁸²			
050	2 · 10 ⁻¹²¹	42 180	60 735	18 271	1 · 10 ⁻⁶⁰⁰²	6 · 10 ⁻⁶⁰⁰²	1 · 10 ⁻⁵⁹⁹⁹	1 · 10 ⁻⁵⁹⁹⁹	1 · 10 ⁻⁵⁹⁹⁹			
060	2 · 10 ⁻¹²¹	46 556	65 014	14 801	2 · 10 ⁻⁷²²⁰	1 · 10 ⁻⁷²²¹	9 · 10 ⁻⁷²¹⁸	9 · 10 ⁻⁷²¹⁸	9 · 10 ⁻⁷²¹⁸			
070	3 · 10 ⁻¹²¹	46 146	65 173	21 369	4 · 10 ⁻⁸⁴³⁹	3 · 10 ⁻⁸⁴⁴⁰	1 · 10 ⁻⁸⁴³⁷	1 · 10 ⁻⁸⁴³⁷	1 · 10 ⁻⁸⁴³⁷			
080	3 · 10 ⁻¹²¹	43 748	65 293	16 353	4 · 10 ⁻⁹⁶⁵⁹	4 · 10 ⁻⁹⁶⁶⁰	5 · 10 ⁻⁹⁶⁵⁷	5 · 10 ⁻⁹⁶⁵⁷	5 · 10 ⁻⁹⁶⁵⁷			
090	4 · 10 ⁻¹²¹	37 447	65 411	8 691	4 · 10 ⁻¹⁰⁸⁷⁹	7 · 10 ⁻¹⁰⁸⁸⁰	1 · 10 ⁻¹⁰⁸⁷⁷	1 · 10 ⁻¹⁰⁸⁷⁷	1 · 10 ⁻¹⁰⁸⁷⁷			
100	4 · 10 ⁻¹²¹	48 561	65 362	19 542	3 · 10 ⁻¹²⁰⁹⁹	1 · 10 ⁻¹²¹⁰⁰	4 · 10 ⁻¹²⁰⁹⁷	4 · 10 ⁻¹²⁰⁹⁷	4 · 10 ⁻¹²⁰⁹⁷			

Knapsack: LLU_Iterative, d=50, factor ≈ 4.0												
dim	time in s				reduction steps				correction steps			
	0	max	min	0	max	min	0	max	min	0	max	min
100	192	832	756	66 072	68 932	63 968	3 1879	32 648	74 123	70 301	966	948
110	1116	1174	1036	85 5456	90 9792	73 8742	37 0709	39 094	1247	1195	2332	2881
120	1470	1540	1396	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	43 282	44 401	14 2118	15 08	15 43	14 86
130	1916	2013	1756	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	49 355	50 414	46 846	11 1617	11 4460	10 5557
140	2438	2611	2247	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	55 570	57 245	53 093	12 5363	12 8838	11 9080
150	3078	3274	2850	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	62 876	64 489	60 526	14 0308	14 4724	13 5837
160	3792	4158	3626	2 · 10 ⁶	2 · 10 ⁶	2 · 10 ⁶	69 021	71 535	67 270	15 4489	16 1223	15 0208
170	4815	5135	4564	2 · 10 ⁶	2 · 10 ⁶	2 · 10 ⁶	77 369	78 03	71 938	17 5730	16 8463	16 0843
180	5826	6149	5433	2 · 10 ⁶	2 · 10 ⁶	2 · 10 ⁶	84 761	86 533	82 583	18 7969	19 2621	18 2535
190	8128	8153	8911	3 · 10 ⁶	3 · 10 ⁶	3 · 10 ⁶	92 812	94 275	90 757	20 4117	20 8295	19 6370
200	8574	9102	8143	3 · 10 ⁶	3 · 10 ⁶	3 · 10 ⁶	100 750	103 404	98 677	22 0692	22 7149	21 5704

Knapsack: LU_Iterative, d=50, factor ≈ 4.0												
dim	0				step packs				exact SP			
	0	max	min	0	max	min	0	max	min	0	max	min
100	122	142	122	122	2881	2332	1925	1925	1925	142	142	142
110	1516	1616	1516	1516	2656	2656	1960	1960	1960	233	233	233
120	194	214	194	194	270	322	244	244	244	289	289	289
130	244	264	244	244	301	338	401	401	401	452	452	452
140	289	309	289	289	408	468	509	509	509	488	488	488
150	332	352	332	332	488	548	592	592	592	488	488	488
160	387	407	387	387	548	607	657	657	657	488	488	488
170	436	456	436	436	607	667	727	727	727	488	488	488
180	484	504	484	484	667	727	787	787	787	488	488	488
190	530	550	530	530	727	787	847	847	847	488	488	488
200	582	602	582	582	787	847	907	907	907	488	488	488

Knapsack: LLL _c , factor ≈ 4.0															
dim	0	time in s	min	max	reduction steps	correction steps	steps	swaps	max	min	0	max	min	step backs	exact SP
100	486	519	451	622660	645927	591967	30477	31444	290441	69301	71328	67068	974	984	965
110	697	736	640	814564	858503	752261	37505	349550	82611	855585	79670	1274	919	1112	729
120	929	984	852	1 · 10 ⁶	943768	42294	43669	40551	95843	99239	91969	1549	1605	1519	894
130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
150	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
160	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
170	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
180	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
190	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
200	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Knapsack: LLL _c , Factor ≈ 4.0															
dim	0	time in s	min	max	reduction steps	correction steps	steps	swaps	max	min	0	max	min	step backs	exact SP
100	3900	4070	3698	588279	611678	55868	29898	30866	28870	69301	71328	67068	408	412	404
110	5429	5705	5047	761500	808336	701256	36769	34195	82611	855585	79670	1274	907	495	809
120	7188	7550	6651	954075	1 · 10 ⁶	873331	41540	42656	350599	95843	99239	91969	609	612	604
130	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0
140	12	12	0	0	0	0	0	0	0	0	0	0	0	0	0
150	15	15	0	0	0	0	0	0	0	0	0	0	0	0	0
160	18	18	0	0	0	0	0	0	0	0	0	0	0	0	0
170	22	22	0	0	0	0	0	0	0	0	0	0	0	0	0
180	26	27	26	0	0	0	0	0	0	0	0	0	0	0	0
190	31	32	30	0	0	0	0	0	0	0	0	0	0	0	0
200	36	36	0	0	0	0	0	0	0	0	0	0	0	0	0
Knapsack: LLL _c , Factor ≈ 4.0															
dim	0	time in s	min	max	reduction steps	correction steps	steps	swaps	max	min	0	max	min	step backs	exact SP
100	2874	30413	26917	573757	597128	543559	29637	31003	28611	69301	71328	67068	193	193	193
110	39586	42534	35866	743686	786922	680241	35237	36204	33845	82611	855585	79670	215	216	194
120	51918	55795	46806	931813	988541	851515	41009	42244	33266	95843	99239	91969	236	236	236
130	65048	70620	59568	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	46709	48496	44732	109265	113447	104613	256	256	256
140	81311	87401	72885	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	52877	55214	49935	128209	116955	360	366	354	399
150	97183	101896	90981	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	58764	60186	56737	137503	140510	133224	434	436	433
160	117720	125432	106331	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	65841	67173	61947	153187	156379	146009	470	470	470
170	1338972	147439	130178	2 · 10 ⁶	2 · 10 ⁶	1 · 10 ⁶	72153	74489	69216	168456	173364	161118	502	502	502
180	161728	173160	154415	2 · 10 ⁶	2 · 10 ⁶	1 · 10 ⁶	78666	80368	77185	185760	187830	179984	533	533	533
190	188975	196902	179233	2 · 10 ⁶	2 · 10 ⁶	2 · 10 ⁶	85598	87400	83607	200099	203214	195754	565	565	565
200	226117	227549	21136	2 · 10 ⁶	3 · 10 ⁶	2 · 10 ⁶	94593	95202	91382	221118	226656	214152	762	764	755

Random-Det: LLL, 400 bits																
dim	time in s			reduction steps			correction steps			swaps			exact SP			defect
	0	max	min	0	max	min	0	max	min	0	max	min	0	max	min	0
010	0	0	0	1186	1516	933	413	502	323	365	552	248	2424	3345	1516	107
020	15	19	10	12249	14864	9391	2313	2743	1749	4519	2376	20550	107	108	3	8 · 10 ⁻⁶
030	104	120	85	556447	62200	50312	6618	7262	5785	11518	12870	9852	114809	133565	996638	106
040	418	488	347	164775	186624	150249	13468	15144	11774	24611	28737	21410	328704	385441	268442	3
050	1217	1394	1061	395594	453319	345772	236662	26494	20943	44752	51518	38185	732466	8776603	623906	2
060	2952	3300	2639	795574	843703	727106	357471	39146	328933	70301	72465	62016	1 · 10 ⁶	1 · 10 ⁶	124	2 · 10 ⁻²
070	6264	6936	5396	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	35746	58948	48914	105560	117731	93567	2 · 10 ⁶	1 · 10 ⁶	106	1 · 10 ⁻³
080	11537	12897	10414	2 · 10 ⁶	2 · 10 ⁶	2 · 10 ⁶	28784	78664	67408	144514	157491	131781	4 · 10 ⁶	5 · 10 ⁶	2	3 · 10 ⁻²
090	20054	21743	18499	3 · 10 ⁶	3 · 10 ⁶	3 · 10 ⁶	50335	11058	88657	190597	203664	176214	6 · 10 ⁶	7 · 10 ⁶	79	1 · 10 ⁻¹
100	32025	34491	29295	5 · 10 ⁶	5 · 10 ⁶	5 · 10 ⁶	117761	125710	109895	236111	254338	216953	9 · 10 ⁶	8 · 10 ⁶	2	1 · 10 ⁻³
Random-Det: Modular ₁ , 400 bits																
dim	0	max	min	0	max	min	0	max	min	0	max	min	0	max	min	0
010	0	0	0	3056	3533	2485	1256	1389	1016	2827	3183	2235	541	1064	31	10 ⁻⁸
020	11	16	10	47786	52698	41989	9493	10202	8821	23134	24817	21621	4824	11794	1985	107
030	75	108	68	235161	252666	206050	27524	29060	254533	65855	68849	593989	17480	358688	9205	106
040	271	290	218	684544	736735	572425	53493	542871	126791	136516	112014	44595	131303	49790	3	4
050	765	805	710	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	91346	95862	84841	215362	225505	204218	78663	105613	40922	2
060	1777	1872	1585	3 · 10 ⁶	3 · 10 ⁶	2 · 10 ⁶	139454	145802	125662	325301	340330	295000	132632	185801	86237	124
070	3508	3665	3196	5 · 10 ⁶	5 · 10 ⁶	5 · 10 ⁶	198292	2208469	181197	456905	4722608	423608	220597	331026	78932	106
080	6417	7274	5665	8 · 10 ⁶	9 · 10 ⁶	8 · 10 ⁶	267272	301463	239135	602700	801701	658865	638385	68031	81	1 · 10 ⁻⁵
090	10187	10805	8884	1 · 10 ⁷	1 · 10 ⁷	1 · 10 ⁷	328833	349554	2865596	757200	801701	658863	436650	739765	762346	721116
100	16420	17214	14896	1 · 10 ⁷	2 · 10 ⁷	1 · 10 ⁷	416084	436650	968491	1 · 10 ⁶	855863	2400 bits	1	1	1	2 · 10 ⁻²
Random-Det: Modular ₂ , 400 bits																
dim	0	max	min	0	max	min	0	max	min	0	max	min	0	max	min	0
010	0	0	0	3056	3533	2485	1256	1389	1016	2827	3183	2235	540	1029	31	10 ⁻⁸
020	11	12	10	47918	53470	41989	9487	10357	8801	23172	24850	21621	4837	11624	1985	107
030	73	77	66	235315	253054	203875	27446	28865	24904	55961	69104	60168	17409	35619	8547	106
040	273	295	224	681442	734201	558584	53116	56558	43407	126342	134057	102052	43016	64184	292220	3
050	760	792	698	1 · 10 ⁶	1 · 10 ⁶	1 · 10 ⁶	90997	94649	83727	215757	224703	1989373	789393	114257	47801	2
060	1746	1827	1600	3 · 10 ⁶	2 · 10 ⁶	1 · 10 ⁶	137235	143003	125235	324347	338874	251586	140017	207146	86766	124
070	3415	3540	3163	5 · 10 ⁶	5 · 10 ⁶	5 · 10 ⁶	192160	199317	178963	455812	469435	422036	220240	350160	82774	106
080	6129	6445	5594	8 · 10 ⁶	9 · 10 ⁶	7 · 10 ⁶	253157	265370	233264	597666	627597	550809	313361	417816	101746	3
090	9961	10640	8733	1 · 10 ⁷	1 · 10 ⁷	1 · 10 ⁷	318876	337980	275851	753651	800459	646008	646008	74977	81	1 · 10 ⁻⁵
100	16189	16878	14801	1 · 10 ⁷	1 · 10 ⁷	1 · 10 ⁷	408896	426773	375457	965134	1 · 10 ⁶	886478	249331	801670	67546	7

Baby-Step Giant-Step Algorithms for Non-uniform Distributions

Simon R. Blackburn^{*1} and Edlyn Teske²

¹ Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, United Kingdom
s.blackburn@rhbnc.ac.uk

² Dept. of Combinatorics and Optimization
Centre for Applied Cryptographic Research
University of Waterloo, Waterloo, ON, N2L 3G1 Canada
eteske@math.uwaterloo.ca

Abstract. The baby-step giant-step algorithm, due to Shanks, may be used to solve the discrete logarithm problem in arbitrary groups. The paper explores a generalisation of this algorithm, where extra baby steps may be computed after carrying out giant steps (thus increasing the giant step size). The paper considers the problem of deciding how many, and when, extra baby steps should be computed so that the expected cost of the generalised algorithm is minimised. When the logarithms are uniformly distributed over an interval of length n , the expected cost of the generalised algorithm is 6% lower than that of Shanks (achieved at the expense of a slightly larger worst case cost). In some situations where logarithms are far from uniformly distributed, any baby-step giant-step algorithm that computes all its baby steps before taking a giant step must have infinite expected cost, but the generalised algorithm has finite expected cost. The results are heuristic, but are supported by evidence from simulations.

1 Introduction

The classic baby-step giant-step algorithm due to Shanks (see, for example, Cohen [1]) makes use of a time-memory trade-off to search an interval of length n for a discrete logarithm using only $O(\sqrt{n})$ operations. In the standard application, we assume that the distribution of answers (logarithms) is uniform over the interval.

This paper considers the following situation. Suppose a baby-step giant-step algorithm is to be used to find a discrete logarithm, and that the logarithm is taken from a known distribution that is not necessarily uniform. (This distribution could be rigorously derived, but it could also be found by experimentation or arise from heuristic results.) How can we minimise the expected number of operations of the algorithm? For which distributions is it a good idea to compute more baby steps after some giant steps have been carried out (so increasing the

^{*} Supported by an E.P.S.R.C. Advanced Fellowship

giant step size)? Our motivation is from recent papers of Stein and Teske [5,6], that give an algorithm to find the divisor class number of a hyperelliptic function field. Their approach is essentially a baby-step giant-step algorithm that searches an interval for a discrete logarithm that is approximately normally distributed. We will comment briefly on this situation in the penultimate section of the paper. However, in general this paper emphasises the study of baby-step giant-step algorithms rather than specific applications.

Our goal is to design a baby-step giant-step algorithm that will return an integer picked from a given distribution using the smallest expected number of operations. We will consider a one sided distribution, where small integer values are more likely to occur than larger values. This can be easily adapted to two sided distributions such as those considered by Stein and Teske — see the penultimate section of the paper.

The theoretical and experimental data both suggest that the best way of computing baby steps depends on the hazard function of the distribution (see the next section for a definition of this concept). For non-increasing distributions on the non-negative integers, we conclude the following. If the distribution has expected value E and increasing hazard rate, then \sqrt{E} baby steps should be computed, and then all the giant steps should be carried out. If, however, the distribution has decreasing hazard rate, extra baby steps should be computed after some giant steps have been taken — see the next section for details.

The arguments in this paper are not rigorous. In particular, we are very cavalier with error terms. Instead, we support our arguments with experimental data.

The remainder of the paper is organised as follows. Section 2 presents the generalisation of the baby-step giant-step algorithm we will consider. Arguments are given as to why this algorithm is optimal. Section 3 lists several distributions and calculates the theoretically optimal algorithm for each of them. Finally, Section 4 presents the simulation results that support the arguments in Section 2.

2 A Baby-Step Giant-Step Algorithm and Its Optimisation

This section analyses a baby-step giant-step algorithm that finds an integer s taken from a distribution on the non-negative integers. Rather than computing all the baby steps at the beginning, the algorithm computes extra baby steps (thus increasing the giant step size) after each unsuccessful giant step. The number of baby steps that are computed is controlled by a function b . The question this paper addresses is: For a given distribution, what choice of b minimises the expected number of operations of the algorithm?

Let b be a monotonic increasing positive-integer valued function on the non-negative integers. In the algorithm below, the variable x holds the smallest integer not checked at any stage; the variable y holds the number of baby steps that have been computed.

1. Set $x = 0$ and $y = 0$.
2. Compute $b(x) - y$ extra baby steps, so that a total of $b(x)$ baby steps have been computed. Set $y = b(x)$.
3. Perform a giant step (of size $b(x)$) to scan the interval $[x, x + b(x) - 1]$. If s lies in this interval, output s and stop.
4. Set $x = x + b(x)$. Return to step 2.

If it is known that $s \leq n$ for some integer n , and if b is the constant function with value $\lceil \sqrt{n} \rceil$, we have the original algorithm of Shanks. If b is any function such that $b(2(i+1) + \frac{1}{2}i(i-1)) = i+2$ for all $i \in \{0, 1, 2, \dots\}$, we have the algorithm recently proposed by Terr [7].

In order to choose a sensible function b for a particular application, we restrict the distributions we consider to those that are non-increasing. More precisely (and to fix notation), let p_0, p_1, \dots be non-negative real numbers such that $\sum_{i=0}^{\infty} p_i = 1$. Assume that $p_i \geq p_{i+1}$ for all i . We consider the distribution associated with the probabilities p_i . Let X be a random variable taking value i with probability p_i , and let E be the expected value of X .

We now define a function b that comes close to minimising the expected number of operations required by the algorithm above. (We note that most of the information encoded in the function b is not used. For example, none of the values $b(1), b(2), \dots, b(b(0)-1)$ affect the algorithm. We assume that these unused values of b are chosen so as to make b a ‘natural’ monotonically increasing function that interpolates the points that matter.) Define a function m , from the non-negative integers to the reals by

$$m(k) = \sqrt{\frac{\sum_{i=k+1}^{\infty} p_i}{p_k}}. \quad (1)$$

If m is monotonic increasing, let b be a function such that $b(k)$ is an integer as close as possible to $m(k)$, subject to $b(k) \geq 1$. If m is monotonic decreasing, let b be a constant function equal to an integer which is approximately \sqrt{E} . If m is not monotonic, then a good choice for b consists of segments that approximate m (since b is monotonic increasing, m must be monotonic increasing on these segments), together with horizontal lines that join these segments. (In this last circumstance, there does not seem to be a simple rule to determine b in general. However, given a distribution, it is not difficult to search for an optimal value of b from functions of this form.)

This choice of the function b is partly justified by simulations; see the final section of the paper. The rest of this section attempts to give a conceptual justification for this choice of b , by analysing a second algorithm. This second algorithm would not be used in practice. However, its expected cost is easier to analyse, and simulations show that the two algorithms have comparable costs. Before carrying out this analysis, we digress to discuss the relationship between the choice of b given above and a concept in statistics known as the hazard rate.

If the distribution does not decay too rapidly, the ratio inside the square root in (1) is well approximated by $(\sum_{i=k}^{\infty} p_i)/p_k$. This quantity is known as the Mills'

ratio, and its reciprocal is known as the hazard rate (or failure rate) at k ; see Johnson, Kotz and Kemp [2, p.111]. If X is the random variable corresponding to the probabilities p_i , the hazard rate at k measures the likelihood of X lying in a small interval beyond k , given that $X \geq k$. (It is used in contexts where the random variable models the likelihood of some machine component failing within a time interval.) We then have an intuitive explanation for the form of the function b above. For if the hazard ratio is high at a point x , and we know that $s \notin [0, x - 1]$, then it is likely that s will occur soon, and so few baby steps should be computed. If the hazard ratio drops as x increases, more baby steps should therefore be computed at each stage, and so b should be an increasing function. Distributions of this type are known as Decreasing Hazard Rate (DHR) distributions; they tend to be distributions with large tails. If the hazard rate increases (so the distribution is an Increasing Hazard Rate (IHR) distribution), the optimal number of baby steps at a point drops. But in this case, since baby step computations cannot be taken back once they are computed, the best we can do is to compute no extra baby steps, and so b is a constant function. This argument gives some reasons why we might expect b to depend on the hazard rate, but does not reveal the precise form of b . We now give an argument that indicates this more precise form.

We briefly describe our second algorithm. Let h be a positive integer. Divide the non-negative integers into intervals I_0, I_1, \dots of length h , where $I_j = [jh, (j+1)h - 1]$. Define $P_j = \sum_{i=jh}^{(j+1)h-1} p_i$; so P_j is the probability that s lies in I_j . To scan each interval I_j , the algorithm computes some baby steps, increasing the giant step size to $b(jh)$ steps. The algorithm then uses $\lceil h/b(jh) \rceil$ giant steps to test whether $s \in I_j$.

We may regard this second algorithm as an approximation to the first, where the number of baby steps is only updated when h integers have been scanned since the last update. For all the approximations we will make to be reasonable, h should be of a moderate size, slightly larger than the square root of the expected value of a random variable taking the value k with probability p_k , say.

The expected number of operations needed to find s using this second algorithm is approximately

$$\sum_{j=0}^{\infty} P_j \left(b(jh) + \left\lceil \frac{h}{b(0)} \right\rceil + \left\lceil \frac{h}{b(h)} \right\rceil + \cdots + \left\lceil \frac{h}{b((j-1)h)} \right\rceil \right).$$

This is a lower bound for the expected number of operations. An upper bound is the same expression with a term $\lceil h/b(jh) \rceil$ added to the sum. A more precise estimate would replace this term with $(E(X|X \in I_j) - jh)/b(jh)$, where the random variable X is such that $\Pr(X = i) = p_i$.

Rearranging this sum, and ignoring the rounding errors caused by requiring the number of giant steps on each interval to be integers, we find that the total expected cost of the algorithm is

$$\sum_{k=0}^{\infty} \left(P_k b(kh) + \frac{h \sum_{j=k+1}^{\infty} P_j}{b(kh)} \right). \quad (2)$$

If we wish to minimise the expected cost of the algorithm, we need to choose the function b so as to minimise the expression (2). Define $m(kh)$ to be that value of $b(kh)$ that minimises the k -th term of (2). So m is a function from the set of non-negative multiples of h to the non-negative integers, and

$$m(kh) = \sqrt{\frac{h}{P_k} \left(\sum_{j=k+1}^{\infty} P_j \right)}, \quad (3)$$

if this expression is an integer, or is one of the two integers closest to this value otherwise. (Note that if $P_k = 0$ then this value is not well defined; but in this case the value of $b(kh)$ does not affect the running time of the algorithm. So when $P_k = 0$, we set $m(kh) = m((k-1)h)$ without loss of generality.)

If the function m defined by (3) is monotonic increasing, setting $b(kh) = m(kh)$ for all k minimises the expected cost of the algorithm. But for many distributions, m is not monotonic increasing and so we cannot use $m(kh)$ to define $b(kh)$ at all values of kh . However, if b is a function that minimises the cost of the algorithm it is not difficult to see that either $b(kh) = m(kh)$, or $b(kh) > m(kh)$ and $b(kh) = b((k-1)h)$, or $b(kh) < m(kh)$ and $b(kh) = b((k+1)h)$. Thus a minimal function b is made up of horizontal line segments, together with segments of the function m defined by (3). In particular, when m is monotonic decreasing (which is the case for a great many sensible probability distributions), the minimising function b is a constant function. Interpreted in algorithmic terms, this means that all the baby steps should be computed at the beginning, before any giant steps are carried out — this is the approach of the original algorithm of Shanks. When b is constant, the value of b giving minimal expected cost is about $\sqrt{E(X)}$, where X is the random variable such that $\Pr(X = i) = p_i$ for all i . To see this, when $b(k) = b$ for some integer b we may write the cost function (2) as

$$\sum_{k=0}^{\infty} \left(P_k b + \frac{h}{b} (P_k + P_{k+1} + \dots) \right) = b + \sum_{k=0}^{\infty} \frac{hk}{b} P_k.$$

This last sum is approximately $E(X)/b$ provided that h is small enough, since the event that s lies in I_k occurs with probability P_k and this event occurring means that s is approximately hk . Since the minimum value of $b+E(X)/b$ occurs when $b = \sqrt{E(X)}$, choosing b to be this value should minimise the expected cost.

3 Example Distributions

This section considers various distributions, and determines a suitable strategy for computing baby steps in each case.

3.1 The Uniform Distribution

Suppose s is uniformly distributed throughout the interval $[0, n - 1]$, so

$$p_i = \begin{cases} \frac{1}{n} & \text{if } i < n, \\ 0 & \text{otherwise.} \end{cases}$$

This is the situation most often considered. In this case, $m(k)$ is approximately $\sqrt{n-k}$ when $k < n$, and $m(k) = 0$ when $k \geq n$. Since m is monotonic decreasing, the optimal choice for b is a constant function approximately equal to $\sqrt{n/2}$. So the original approach of Shanks, computing all the baby steps first, is the best possible here. (However, Shanks' version of computing approximately \sqrt{n} baby steps yields a worse average-case running time but a better worst-case running time.)

3.2 The One Sided Normal Distribution

Suppose that s follows a discrete approximation to one side of a normal distribution of mean 0 and standard deviation σ . So

$$p_i \simeq \frac{2}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}i^2}.$$

The normal distribution is an IHR distribution (see, for example, Kececioglu [3, p.362], and so the optimal value for b should be a constant function with value $\sqrt{\sqrt{2}\sigma/\sqrt{\pi}}$.

3.3 The Discrete Pareto Distribution

Let $p_i = c/(i+1)^d$, where $d > 1$ and where c is chosen so that $\sum_{i=0}^{\infty} p_i = 1$, i.e. $c = 1/\zeta(d)$, where $\zeta(\cdot)$ denotes the Riemann zeta function. (This distribution is also referred to as the Zipf distribution, or the Riemann zeta distribution.)

Then the function $m(k)$ defined by (1) is approximately $\sqrt{k}/(d-1)$. Since this function is monotonic increasing, we set $b(k) = \max\{1, [m(k)]\}$ in this case. For this choice of $b(k)$, our algorithm becomes similar to an algorithm recently proposed by Terr [7]. Note that when $d \leq 2$, an algorithm that computes all its baby steps at the beginning has an infinite expected cost, while for the algorithm with $b(k) = m(k)$ has infinite expected cost only when $1 < d < 3/2$.

3.4 The Weibull Distribution

The discrete Weibull distribution is approximately given by

$$p_i = \frac{\beta}{\eta} \left(\frac{i}{\eta} \right)^{\beta-1} e^{-(i/\eta)^\beta},$$

where the positive real numbers β and η are respectively the shape parameter and the scale parameter. We only consider the case when $0 < \beta \leq 1$ — the case when $\beta = 1$ is the geometric distribution (see the next subsection) and when $\beta > 1$, the distribution is not decreasing (the p_i rapidly climb to a maximum value, before falling gently towards zero).

The hazard function of the distribution is approximately $\lambda(k) = \beta/\eta \cdot (k/\eta)^{\beta-1}$, which is strictly decreasing for $0 < \beta < 1$. The expected value $E(X)$ is approximately $\eta \Gamma(1/\beta + 1)$, where Γ denotes the gamma function. Now, $m(k)$ is approximately $1/\sqrt{\lambda(k)}$ so we set $b(k) = \max\{1, m(k)\}$ for $k \geq 1$, and $b(0) = b(1)$.

3.5 The Geometric Distribution

Let $p_i = qp^i$, where $p + q = 1$. The function m defined by (1) is approximately $\sqrt{p/q}$, a constant function. Note that $E(X) = p/q$ in this case, and so the value for b obtained by using the function m is equal to the value of b that is obtained by minimising the expected cost over all constant functions.

3.6 The Split Uniform Distribution

Suppose that the interval $[0, n - 1]$ is divided into two parts $[0, \ell - 1]$ and $[\ell, n - 1]$ in each of which s is uniformly distributed, i.e.

$$p_i = \begin{cases} p_A & \text{if } i < \ell , \\ p_B & \text{if } \ell \leq i < n , \\ 0 & \text{otherwise ,} \end{cases}$$

where p_A , p_B and ℓ are such that $p_A \geq p_B$ and $\ell p_A + (n - \ell)p_B = 1$, i.e., $p_B = (1 - \ell p_A)/(n - \ell)$. Then $m(k) = \sqrt{1/p_A - (k + 1)}$ if $k < \ell$ and $m(k) = \sqrt{n - (k + 1)}$ if $\ell \leq k < n$, so that m is neither decreasing nor increasing. Since m is decreasing on the intervals $[0, \ell - 1]$ and $[\ell, n - 1]$, the function b should be constant on these two intervals. Thus b is determined by $b(0)$ and $b(\ell)$. So a good choice of b is computed by minimising the expected cost over all choices of $b(0)$ and $b(\ell)$; in other words, by minimising

$$\left(b(0) + \frac{\ell}{2b(0)} \right) \ell p_A + \left(b(\ell) + \frac{\ell}{b(0)} + \frac{n - \ell}{2b(\ell)} \right) (n - \ell) p_B .$$

Hence, the best values for $b(0)$ and $b(\ell)$ should be approximately

$$b(0) = \sqrt{\frac{1}{p_A} \left(\frac{1}{2} \ell p_A + (n - \ell) p_B \right)} \text{ and } b(\ell) = \sqrt{\frac{1}{2} (n - \ell)} ,$$

which yields an increasing function if $p_A > 2/n$. If $p_A \leq 2/n$, the cost function is minimised by a constant function, with value $\sqrt{E(X)}$, where

$$E(X) = \frac{\ell^2}{2} p_A + \frac{n^2 - \ell^2}{2} p_B .$$

3.7 Two Sided Distributions

The distributions we have considered have all been non-increasing, starting at 0. We may easily modify our treatment to deal with two sided distributions such as discrete approximations to the normal distribution. In this case, our algorithm should start at the peak value of the distribution, making giant steps in both directions away from the mean. In the case of a symmetrical distribution about 0, the cost function may be approximated by

$$\sum_{k=0}^{\infty} P_k b(kh) + \frac{2h \sum_{j=k+1}^{\infty} P_j}{b(kh)} . \quad (4)$$

where here P_k is the probability that $s \in I_k \cup -I_k$. This changed cost function leads to an altered function $m(k)$, that differs by a factor of $\sqrt{2}$ from the original. The optimal value for a constant function b is equal to $\sqrt{2E(|X|)}$.

To take a specific example, if s is approximately normally distributed with mean 0 and standard deviation σ , we find that the optimal choice of b is a constant function with value $\sqrt{2\sqrt{2}\sigma/\sqrt{\pi}}$.

4 Experimental Results

We presented two algorithms in Section 2. The first algorithm, the ‘practical’ algorithm, has a running time that is difficult to analyse precisely. The second algorithm, the ‘theoretical’ algorithm, was used to derive an optimal choice for the baby-step function b . It is important to verify experimentally that the running times of the two algorithms are comparable when a theoretically optimal choice of b is used, and that the optimal value for b predicted by the theory accords well with practice. This section contains the experimental results we have obtained.

Subsection 4.1 gives the mean cost of the practical and theoretical algorithms, when s is taken from some of the distributions considered in Section 3. Their costs are found to be comparable. Baby step functions corresponding to the algorithms of Shanks and Terr are also included for the purpose of comparison.

Subsection 4.2 attempts to find the best choice of the function b experimentally. The optimal choice for b via experiment is shown to agree well with the choice predicted by the theory.

For all our experiments, we used the computer algebra system LiDIA [4].

4.1 Testing the Theoretically Optimal Baby-Step Functions

This subsection compares the mean costs of the practical and theoretical algorithms on a range of distributions, where the baby step function b is taken to be the theoretically best choice. Costs of the algorithms of Shanks and of Terr are also included, by considering appropriate baby step functions b .

To this end, we conducted the following experiment. We considered approximately N values of s , taken from various distributions p_0, p_1, \dots, p_{n-1} on $[0, n-1]$. In our experiment, we took $n = 100000$ and $N = 2 \cdot 10^9$. For each distribution, we carried out the following steps:

1. *Simulate the probability distribution on $[0, n-1]$:*

For $0 \leq i < n$, let s_i be the nearest integer to $p_i N$. Let $\hat{N} = \sum_{i=0}^{n-1} s_i$, and let $\hat{p}_i = s_i/\hat{N}$. Compute the expected value $E(s) = \sum_{i=0}^{n-1} i\hat{p}_i$.

2. *Compute the theoretically optimal baby-step function:*

If the distribution is IHR, let $b_{\text{opt}}(0)$ be the closest integer to $\sqrt{E(s)}$, and let $b_{\text{opt}}(k) = b_{\text{opt}}(k-1)$ for $1 \leq k < n$.

If the distribution is DHR, use (1) with p_i, p_k replaced by \hat{p}_i, \hat{p}_k to find $m(k)$ ($0 \leq k < n$) and let $b_{\text{opt}}(k)$ be the integer closest to $m(k)$ subject to the

conditions that $b_{\text{opt}}(k) \geq 1$ and that b_{opt} is not decreasing. In the case of the split uniform distribution, choose the function b as given in Subsection 3.6. For the explicit baby-step functions see further below.

3. *Compute Shanks-type baby-step functions:*

Define $b_{S,n}(k) = \lceil \sqrt{n} \rceil$ for $0 \leq k < n$. This is the baby-step function which is canonically used in baby-step giant-step algorithms. This function does not depend on the probability distribution.

If the distribution is not IHR, we also define the function $b_{S,E}$ as a constant function with value $\sqrt{E(s)}$. This represents the best choice of baby-step function, if one is restricted to computing all the baby steps before a giant step is carried out; if the distribution is IHR, this function is identical with b_{opt} .

4. *Compute Terr's baby-step function:*

Let $b_T(k) = 2$ for $0 \leq k < 4$ and, for $k \geq 4$, let $b_T(k) = j + 2$ where $j = j(k)$ is the uniquely determined integer such that $2(j+1) + \frac{1}{2}j(j-1) \leq k < 2(j+2) + \frac{1}{2}j(j+1)$. The function $b_T(k)$ essentially grows as $\sqrt{2k}$. For example, we have the following values:

k	0	4	7	11	16	22	...	106	121	...	1036	1082	...	10012	10154	...	99682
$b_T(k)$	2	3	4	5	6	7	...	15	16	...	46	47	...	142	143	...	447

The practical algorithm with this choice of baby-step function is essentially identical to the algorithm of Terr.

5. *For each baby-step function above, determine the mean cost of the practical algorithm:*

For each $i \in [0, n - 1]$ with $s_i \neq 0$ we count the numbers of baby steps and giant steps needed to scan the interval $[0, i]$ using the practical algorithm, and multiply the respective results by s_i . We add these results up for all i and divide the respective sums by \hat{N} . This gives the average numbers of baby steps and giant steps. The mean cost is just the sum of these two numbers.

6. *For each baby-step function above, determine the mean cost of the theoretical algorithm:*

Let h be the integer nearest to $\sqrt{E(s)}$. Then we proceed analogously to the practical algorithm. Notice that, however, we do this only for Terr's baby-step function and, if the distribution is not IHR, for the theoretically optimal function; for all Shanks-type functions, the theoretical and the practical algorithms are identical.

The Explicit Probability Distributions and their Theoretically Optimal Baby-Step Functions.

Recall that $n = 100000$ in all our experiments.

The *uniform distribution* on $[0, n - 1]$ has expected value $n/2$, so the theoretically optimal baby-step function is given by $b_{\text{opt}}(k) = 224$ for all k .

When considering the *one-sided normal distribution*, we work with $\sigma = 25000$. On $[0, n - 1]$ we find that $E(s) = 19941.4$, so the theoretically optimal baby-step function is given by $b_{\text{opt}}(k) = 141$ for all k .

Table 1. Testing the b 's. $n = 100000$.

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pr. = Th.	optimal	224.000	224.712	448.712
Pr. = Th.	Shanks, 317	317.000	159.225	476.225
Pract.	Terr	299.130	299.138	598.269
Theor. ($h = 224$)	Terr	297.908	396.789	694.698
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pr. = Th.	optimal	141.000	142.920	283.920
Pr. = Th.	Shanks, 317	317.000	64.395	381.395
Pract.	Terr	184.805	184.812	369.617
Theor. ($h = 141$)	Terr	183.511	244.451	427.962
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.363	25.675	53.038
Theor. ($h = 50$)	optimal	25.441	31.739	57.180
Pr. = Th.	Shanks, 50	50.000	50.164	100.164
Pr. = Th.	Shanks, 317	317.000	8.777	325.777
Pract.	Terr	29.016	28.858	57.875
Theor. ($h = 50$)	Terr	27.371	36.198	63.569
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	optimal	107.772	104.297	212.069
Theor. ($h = 114$)	optimal	105.769	111.365	217.135
Pr. = Th.	Shanks, 114	114.000	114.427	228.427
Pr. = Th.	Shanks, 317	317.000	42.009	359.009
Pract.	Terr	123.536	123.560	247.096
Theor. ($h = 114$)	Terr	121.468	166.963	288.432
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pr. = Th.	optimal	100.000	101.435	201.435
Pr. = Th.	Shanks, 317	317.000	32.999	349.999
Pract.	Terr	126.283	126.295	252.579
Theor. ($h = 100$)	Terr	124.918	166.920	291.838
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	optimal	137.952	139.861	277.813
Theor. ($h = 141$)	optimal	137.598	139.939	277.537
Pr. = Th.	Shanks, 141	141.000	143.338	284.338
Pr. = Th.	Shanks, 317	317.000	64.579	381.579
Pract.	Terr	175.519	175.528	351.047
Theor. ($h = 141$)	Terr	174.181	235.003	409.184

For the *Pareto distribution*, we choose $d = 1.2$, which yields $E(s) = 2453.9$. The theoretically optimal function $b_{\text{opt}}(k)$ looks as follows:

$$\begin{array}{c|cccccccccccccccc}
k & 0 & 2 & 3 & 5 & 8 & \dots & 107 & 119 & \dots & 1006 & 1048 & \dots & 10036 & 10263 & \dots & 19546 & \dots & 39172 \\
\hline b(k) & 3 & 4 & 5 & 6 & 7 & \dots & 21 & 22 & \dots & 56 & 57 & \dots & 137 & 138 & \dots & 166 & \dots & 184
\end{array}.$$

Here, b_{opt} is a step function, and we always tabulate the least value k for which b_{opt} assumes the indicated value. In particular, for all $39172 \leq k < n$ we have $b_{\text{opt}}(k) = 184$.

Table 2. Theoretical versus practical algorithm: varying h .

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pract.	Terr	299.130	299.138	598.269
Theor. ($h = 224$)	Terr	297.908	396.789	694.698
Theor. ($h = 100$)	Terr	297.812	340.653	638.465
Theor. ($h = 50$)	Terr	297.836	318.762	616.598
Theor. ($h = 20$)	Terr	297.924	306.895	604.819
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pract.	Terr	184.805	184.812	369.617
Theor. ($h = 141$)	Terr	183.511	244.451	427.962
Theor. ($h = 100$)	Terr	183.469	225.983	409.453
Theor. ($h = 50$)	Terr	183.428	204.192	387.621
Theor. ($h = 20$)	Terr	183.505	192.415	375.920
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.363	25.675	53.038
Theor. ($h = 50$)	optimal	25.441	31.739	57.180
Theor. ($h = 20$)	optimal	26.396	28.750	55.146
Pract.	Terr	29.016	28.858	57.875
Theor. ($h = 50$)	Terr	27.371	36.198	63.569
Theor. ($h = 20$)	Terr	27.955	31.885	59.840
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	Terr	123.536	123.560	247.096
Theor. ($h = 114$)	Terr	121.468	166.963	288.432
Theor. ($h = 100$)	Terr	121.597	161.398	282.995
Theor. ($h = 50$)	Terr	122.011	141.490	263.501
Theor. ($h = 20$)	Terr	122.221	130.488	252.710
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pract.	Terr	126.283	126.295	252.579
Theor. ($h = 100$)	Terr	124.918	166.920	291.838
Theor. ($h = 50$)	Terr	124.925	145.333	270.259
Theor. ($h = 20$)	Terr	124.931	133.676	258.607
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	Terr	175.519	175.528	351.047
Theor. ($h = 141$)	Terr	174.181	235.003	409.184
Theor. ($h = 100$)	Terr	174.224	216.587	390.812
Theor. ($h = 50$)	Terr	174.127	194.828	368.955
Theor. ($h = 20$)	Terr	174.189	183.076	357.265

For the *Weibull distribution*, we choose $\beta = 0.5$ and $\eta = 10000$. We find that $E(s) = 12882.4$, so the theoretically optimal function $b_{\text{opt}}(k)$ is a step function that takes the following values (tabulating as before):

$$\begin{array}{c|ccccccccccccc}
k & | & 0 & 2 & 3 & 4 & 5 & 6 & \dots & 104 & 113 & \dots & 1045 & 1100 & \dots & 10340 & 10702 & \dots & 35884 \\
\hline b(k) & | & 14 & 17 & 19 & 20 & 21 & 22 & \dots & 45 & 46 & \dots & 79 & 80 & \dots & 135 & 136 & \dots & 166
\end{array} .$$

(In particular, $b_{\text{opt}}(k) = 166$ when $35884 \leq k < n$.)

Table 3. Testing the b 's. $n = 100000$. With shortcut

	b	aver. # bs	aver. # gs	aver. # (bs+gs)
Uniform distribution: $E(s) = 49999.5$				
Pr. = Th.	optimal	223.750	223.712	447.462
Pr. = Th.	Shanks, 317	316.499	158.225	474.725
Pract.	Terr	299.130	298.138	597.269
Theor. ($h = 224$)	Terr	297.908	395.789	693.698
One sided normal distribution: $\sigma = 25000$, $E(s) = 19941.4$				
Pr. = Th.	optimal	140.678	141.920	282.599
Pr. = Th.	Shanks, 317	315.386	63.395	378.782
Pract.	Terr	184.805	183.812	368.617
Theor. ($h = 141$)	Terr	183.511	243.451	426.962
Pareto distribution: $d = 1.2$, $E(s) = 2453.9$				
Pract.	optimal	27.017	25.881	52.898
Theor. ($h = 50$)	optimal	24.963	30.739	55.702
Pr. = Th.	Shanks, 50	23.013	49.164	72.177
Pr. = Th.	Shanks, 317	91.717	7.777	99.494
Pract.	Terr	28.820	27.858	56.678
Theor. ($h = 50$)	Terr	27.175	35.198	62.373
Weibull distribution: $\beta = 0.5$, $\eta = 10000$, $E(s) = 12882.4$				
Pract.	optimal	107.756	105.219	212.975
Theor. ($h = 114$)	optimal	105.544	110.365	215.909
Pr. = Th.	Shanks, 114	106.764	113.427	220.191
Pr. = Th.	Shanks, 317	282.521	41.009	323.53
Pract.	Terr	123.536	122.56	246.096
Theor. ($h = 114$)	Terr	121.468	165.963	287.432
Geometric distribution: $\sigma = 0.9999$, $E(s) = 9994.5$				
Pr. = Th.	optimal	99.502	100.435	199.937
Pr. = Th.	Shanks, 317	312.029	31.999	344.028
Pract.	Terr	126.283	125.295	251.579
Theor. ($h = 100$)	Terr	124.918	165.920	290.838
Split uniform distribution: $\ell = 20000$, $p_A = 1/25000$; $E(s) = 19999.5$				
Pract.	optimal	137.657	138.861	276.518
Theor. ($h = 141$)	optimal	137.303	138.939	276.242
Pr. = Th.	Shanks, 141	140.605	142.338	282.943
Pr. = Th.	Shanks, 317	314.996	63.579	378.575
Pract.	Terr	175.518	174.528	350.047
Theor. ($h = 141$)	Terr	174.181	234.003	408.184

For the *geometric distribution*, we work with $p = 0.9999$, and we find that $E(s) = 9994.5$. We calculate that the theoretically optimal baby-step function is defined by $b_{\text{opt}}(k) = 100$ for all k .

Finally, in the case of the *split uniform distribution*, we work with $\ell = 20000$, and $p_A = 1/25000$. Then $E(s) = 19999.5$ and the optimal baby-step function is such that $b_{\text{opt}}(k) = 122$ for $0 \leq k < 20000$ and $b_{\text{opt}}(k) = 200$ for $20000 \leq k < n$.

Notice that in all distributions above, the parameters have been chosen so that $\max\{i ; s_i \neq 0\} = n - 1$.

The results for this experiment are shown in Table 1. We see that in all cases, we get the best performance for both the practical and the theoretical algorithm if we use b_{opt} . In particular, the differences between the data for b_{opt} and for $b_{S,317}$ are quite impressive, especially in the case of the Pareto, the Weibull, the geometric and the one-sided normal distributions. While $b_{S,E}$ still yields an acceptable or good performance, this is not the case for the Pareto distribution, where the use of a non-constant function b seems to be particularly practical.

Although the theoretical and practical algorithms have comparable costs for a theoretically optimal choice of b , we notice a discrepancy between the performances of other baby-step functions. This phenomenon is particularly striking in the case of Terr's function, and is can be explained as follows: To search the interval $[0, h - 1]$, we use giant steps of length $b_T(0) = 2$ in the theoretical algorithm. Thus, to find x in $[0, h - 1]$, we use two baby steps and $x/2$ giant steps. To search the whole interval $[0, h - 1]$, we use two baby steps and $(h - 1)/2$ giant steps of length two. Only after that, the giant step size is increased to $b_T(h)$, etcetc. This leads to a considerably increased number of giant steps, compared with the practical algorithm where the baby step size is adjusted after each giant step. The effect of this can be reduced if we decrease the size of h , as the results in Table 2 show.

Nevertheless, the theoretically optimal baby-step functions still yield the best performances, both for the theoretical and the practical algorithms.

The Shortcut. If equality checks are cheap, what is the case in the majority of applications, the baby-step giant-step method usually is applied with a “shortcut”: Instead of computing all $b(0)$ baby steps first and then using the first giant step to check whether $i \in [0, b(0) - 1]$, one checks for each newly computed baby step whether i has been found already. This means that one finds all $i < b(0)$ by performing just $i+1$ baby steps and equality checks, and no giant step. This technique is particularly favourable for all Shanks-type functions. To check whether our theoretically optimal baby-step functions still give the best performances, we have implemented the shortcut in both the theoretical and practical algorithms. The results are shown in Table 3, where we see that indeed, the performance of $b_{S,E}$ and $b_{S,317}$ improves, where the most dramatic improvement we observe in the case of the Pareto distribution.

4.2 Experimental Finding of Best Baby-Step Functions

In this section, we try to find best baby-step functions b experimentally. For this, we work with the same probability distributions as above, and we restrict ourselves to monotonically increasing step functions b that are constant on the five intervals $[j \cdot 20000, (j + 1) \cdot 20000[$, $0 \leq j \leq 4$. We use the practical algorithm with the shortcut; by doing this, we risk some discrepancy with our theoretically optimal function, but this is the algorithm that would be used in practice, after all.

Table 4. Optimal step functions (5 steps). With shortcut.

Distribution	$b(0)$	$b(20000)$	$b(40000)$	$b(60000)$	$b(80000)$	aver. #(bs+gs)
Uniform	227	228	228	228	228	447.179
	226	227	227	227	227	447.189
	225	226	226	226	226	447.198
	231	231	231	231	231	447.204
	224	225	225	225	225	447.205
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	→ 224	224	224	224	224	447.249
One-s. normal	142	142	142	142	142	282.611
	→ 141	141	141	141	141	282.612
	142	142	142	142	143	282.612
	141	141	141	141	142	282.613
	142	142	142	142	144	282.613
	→					
Pareto	71	167	167	167	167	52.898
	71	166	166	166	166	59.910
	71	165	165	165	165	59.910
	71	164	164	164	164	59.911
	72	167	167	167	167	59.912
	→					
Weibull	106	155	155	155	155	212.975
	107	156	156	156	156	216.734
	107	153	153	153	153	216.738
	106	156	156	156	156	216.740
	107	154	155	155	155	216.741
	→					
Geometric	100	100	100	100	100	199.946
	100	100	100	100	102	199.947
	100	100	100	101	101	199.947
	100	100	100	100	103	199.947
	100	100	100	101	102	199.947
	→					
Split uniform	124	194	194	194	194	276.265
	124	195	195	195	195	276.265
	124	196	196	196	196	276.265
	124	197	197	197	197	276.266
	124	198	198	198	198	276.267
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	→ 122	200	200	200	200	276.422

For each distribution, we proceed in several rounds, where with each round we converge on a good step function. In each round, we allow seven different values for the steps, which gives 462 distinct increasing step functions with five steps, and we keep track of those 20 combinations of values which yield the 20 lowest average costs. (The costs are computed in the same way as in the previous section.) In the beginning the seven values are uniformly spread over the interval $[0, \max\{\sqrt{n}, 2\sqrt{E(s)}\}]$, and based upon the best values of the previous round, we choose the seven values for the next round, so that eventually we end up with

what we hope are the best choice of values. A selection of our results – the best 5 choices each together with the respective average costs – is shown in Table 4. There, an arrow marks the theoretically optimal function; since the optimal functions for the Pareto and the Weibull distributions can not be written in that form, we simply copied the corresponding performance data for the optimal functions from Table 3. We see that in the case of the one-sided normal and the geometric distributions, the best step function and the theoretically optimal function (almost) coincide. In the case of the uniform and the split uniform distributions, these two function differ slightly. However, the difference in the average costs for the best step function and the theoretically optimal function is very small! Even in the case of the Pareto and the Weibull distributions, where the theoretically optimal function is much more complex, the best step functions yield average costs that are only slightly higher. Finally, it may be interesting to note that for the Pareto distribution from Section 4.1 we have $b_{\text{opt}}(0) = 3$, $b_{\text{opt}}(20000) = 166$ and $b_{\text{opt}}(40000) = b_{\text{opt}}(60000) = b_{\text{opt}}(80000) = 184$, while for the Weibull distribution we have $b_{\text{opt}}(0) = 14$, $b_{\text{opt}}(20000) = 153$ and $b_{\text{opt}}(40000) = b_{\text{opt}}(60000) = b_{\text{opt}}(80000) = 166$.

5 Conclusion

We have examined, in theory and practice, the question of which strategy of computing the baby step set in the baby-step giant-step method yields the best average performance, given the probability distribution of the solution. Our results can be used to considerably speed up the baby-step giant-step method, and to save storage space as well.

In particular, our findings show that the original method of computing \sqrt{n} baby steps at the very beginning, and no other baby steps later (where n is the length of the interval in which the solution lies) is not optimal for any of the standard distributions, even the uniform distribution. Here, optimality is measured in terms of minimising the expected number of operations. Indeed, if the probability distribution of the solution has increasing hazard rate (which is the case for the uniform distribution, the one-sided normal distribution and the geometric distribution) and expected value E , the best average performance is obtained when about \sqrt{E} baby steps are computed first, and then giant steps are computed until the solution is found. In particular, this means that for IHR distributions, the only information needed to execute the optimal strategy is the expected value. On the other hand, if the distribution has decreasing hazard rate (such as the Pareto distribution or the Weibull distribution), the optimal strategy consists in computing some baby steps, then some giant steps, then some more baby steps and so on, where the exact optimal number of steps depends on the whole sequence (p_i) of probabilities. However, if this information is not available, applying the same strategy as for IHR distributions still gives much better results than the original version. Finally, it is worth noting that Terr's algorithm is close to optimal for the Pareto distribution.

Acknowledgments

Many thanks to Sean Murphy, Andreas Stein and Albyn Jones for very useful comments during the writing of this paper.

References

1. H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer, Berlin, 1993.
2. N.L. Johnson, S. Kotz and A.W. Kemp, *Univariate discrete distributions*, Wiley Interscience, 1992.
3. D.B. Kececioglu, *Reliability Engineering Handbook*, vol.1, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
4. LiDIA Group, *LiDIA - A library for computational number theory, Version 1.3*, Technische Universität Darmstadt, Darmstadt, Germany, 1997. Available from <http://www.informatik.tu-darmstadt.de/TI/LiDIA>.
5. A. Stein and E. Teske, ‘Explicit bounds and heuristics on class numbers in hyperelliptic function fields’, University of Waterloo, Centre for Applied Cryptographic Research, Technical Report, CORR 99-26.
6. A. Stein and E. Teske, ‘Optimized baby step-giant step methods for hyperelliptic function fields’, *preprint*.
7. C.D. Terr, ‘A modification of Shanks’ baby-step giant-step algorithm’, *Math. Comp.*, posted on March 4, 1999, PII: S 0025-5718(99)01141-2 (to appear in print).

On Powers as Sums of Two Cubes

Nils Bruin*

Utrecht University,
PO box 80010, 3508 TA Utrecht, The Netherlands

Abstract. In a paper of Kraus, it is proved that $x^3 + y^3 = z^p$ for $p \geq 17$ has only trivial primitive solutions, provided that p satisfies a relatively mild and easily tested condition. In this article we prove that the primitive solutions of $x^3 + y^3 = z^p$ with $p = 4, 5, 7, 11, 13$, correspond to rational points on hyperelliptic curves with Jacobians of relatively small rank. Consequently, Chabauty methods may be applied to try to find all rational points. We do this for $p = 4, 5$, thus proving that $x^3 + y^3 = z^4$ and $x^3 + y^3 = z^5$ have only trivial primitive solutions. In the process we meet a Jacobian of a curve that has more 6-torsion at any prime of good reduction than it has globally. Furthermore, some pointers are given to computational aids for applying Chabauty methods.

1 Introduction

In this paper we consider for given $p = 2, 3, \dots$ the Diophantine equation

$$x^3 + y^3 = z^p, \quad x, y, z \in \mathbb{Z}, \quad \gcd(x, y, z) = 1. \quad (1)$$

To emphasise that we look at solutions with $\gcd(x, y, z) = 1$, we refer to such solutions as *primitive solutions*.

First, we review what is known. This equation is a special case of the generalised Fermat equation $x^r + y^s = z^t$. For fixed exponent triples r, s, t , a theorem of Darmon and Granville (see [8]) gives a classification of what the solution set looks like. If we apply their result to our equation, we get that for each $p \geq 4$, $x^3 + y^3 = z^p$ has only finitely many primitive solutions. If we assume the ABC-conjecture (see [23]), then we even get that for p big enough, $x^3 + y^3 = z^p$ only has the trivial primitive solutions, i.e. with $xyz = 0$. This has led (together with the lack of counterexamples) to the very bold conjecture

Conjecture 1 (Tijdeman, Zagier, Beal Prize Problem) *Let x, y, z, r, s, t be positive integers with $r, s, t > 2$. If $x^r + y^s = z^t$ then x, y, z have a factor in common.*

which even has a reward attached to its resolution (see [13]).

For $p = 2$, the result by Darmon and Granville does not give information. In fact, there are infinitely many primitive solutions to $x^3 + y^3 = z^2$. A paper by Beukers (see [2]) guarantees that these solutions can be finitely parametrised. That means that one can give a finite number of polynomial solutions to $x^3 + y^3 =$

* funded by NWO grant “Groot project getaltheorie”.

z^2 such that each primitive solution can be obtained from one of the polynomial solutions by specialisation. In Lemma 1 we give these parametrisations.

For $p = 3$, the Darmon and Granville result is also inconclusive. However, Euler and maybe even Fermat already proved that the equation $x^3 + y^3 = z^3$ has only trivial rational solutions.

For $p \geq 4$, Darmon and Granville predict that there are only finitely many primitive solutions and it is generally believed that they are all trivial. Once we verify this for $p = 4$, it suffices to prove triviality only for prime p , since any composite $p \geq 4$ is either divisible by 4 or by a prime number ≥ 3 . This justifies that the exponent is tendentiously designated by p , hinting at primality. In this light, one may consider 4 as a composite number with primal tendencies.

For $17 \leq p < 10000$, Kraus (see [12]) has proved that there are no nontrivial primitive solutions, assuming the Taniyama-Weil conjecture that is now considered proved by many. His proof resembles Frey's and Ribet's construction of a nonmodular elliptic curve assuming the existence of a nontrivial solution. For this, however, he needs the existence of a prime number with certain properties. While it seems plausible that such a prime exists for any p , Kraus failed at proving so. Therefore, he checked the condition for all prime numbers in the given range individually.

It is the purpose of this paper to sketch how the cases with $p < 17$ can be dealt with and carry out the procedure in a couple of cases. We will prove

Theorem 1 *Integer solutions to $x^3 + y^3 = z^4$ with $xyz \neq 0$ have $\gcd(x, y, z) > 1$.*

Theorem 2 *Integer solutions to $x^3 + y^3 = z^5$ with $xyz \neq 0$ have $\gcd(x, y, z) > 1$.*

Alternatively, one may try to extend the method of Kraus to smaller p . In his paper, he assumes there is a nontrivial solution and constructs a non-CM elliptic curve from it. He then analyses the Galois-representation on the p torsion and concludes it is not modular. One step consists of showing that the representation should be irreducible. Since the curve constructed by Kraus has a rational 2-torsion point, the curve would correspond to a rational point on the modular curve $X_0(2p)$ if the representation were reducible. Following [9], such curves are either singular or CM for prime $p \geq 7$. The curve $X_0(10)$, however, has genus 0 and has infinitely many rational points. It may be possible to prove irreducibility in this case using other arguments, however. Although the details are not (yet) available in the literature, it seems doable to extend Kraus's methods to $p = 5, 7, 11, 13$ (see [11]). The number 4 still has enough composite features to completely break down Kraus's method for $p = 4$, however.

2 Parametrising Curves; Chabauty Methods

If we use the word *curve*, we mean a smooth projective geometrically irreducible variety of dimension 1. We will often work with singular, planar models of these curves. Some of these models will have singularities at their unique point at

infinity. If the smooth curve has only one point there, we will refer to that as ∞ and if it has two, we call them ∞^+ and ∞^- (arbitrarily but fixed).

In this section, we will construct hyperelliptic curves over \mathbb{Q} such that a primitive solution of $x^3 + y^3 = z^p$ corresponds to a rational point on one of the curves. Furthermore, we will give an estimate for the Mordell-Weil rank of the Jacobians of these curves and sketch how one may proceed in finding all rational points on these curves. The sketched method is not an algorithm, but in practice these methods often work.

Suppose that we have $x, y, z \in \mathbb{Z}$ with $\gcd(x, y, z) = 1$ and $x^3 + y^3 = z^p$. Note that $\gcd(x + y, x^2 - xy + y^2) \mid 3$ (an elementary resultant computation). Therefore, we have $t \in \{-1, 0, 1\}$ and $z_1, z_2 \in \mathbb{Q}$ such that

$$\begin{aligned} x + y &= 3^t z_1^p, \\ x^2 - xy + y^2 &= 3^{-t} z_2^p, \\ z &= z_1 z_2. \end{aligned}$$

We solve for y in the first equation and substitute the value in the second equation and divide by z_1^{2p} . This gives an equation that is of degree p in $\frac{z_2}{z_1^2}$ and quadratic in $\frac{x}{z_1^p}$. For each solution and given t , this gives us a rational point on a fixed curve $\mathcal{C}_{p,t}$ in the following way.

t	X	Y	$\mathcal{C}_{p,t}$	$A_{p,t}$
-1	$\frac{z_2}{z_1^2}$	$54 \frac{x}{z_1^p} + 9$	$Y^2 = 324X^p - 3$	$-2^{2p-2}3^{2p-3}$
0	$\frac{z_2}{z_1^2}$	$6 \frac{x}{z_1^p} + 3$	$Y^2 = 12X^p - 3$	$-2^{2p-2}3^p$
1	$\frac{z_2}{z_1^2}$	$6 \frac{x}{z_1^p} + 9$	$Y^2 = 4X^p - 27$	$-2^{2p-2}3^3$

For odd $p \geq 5$, the curves $\mathcal{C}_{p,t}$ are of genus $(p-1)/2$ and also have a model of the form $Y^2 = X^p + A_{p,t}$. By Faltings' theorem, we have that the number of rational points on a curve of genus ≥ 2 is finite. Finding those points or, if you have them, proving that the list of points is complete, is a different matter. Faltings' proof is not constructive, so it is of little help.

There is an earlier, partial proof by Chabauty (see [6]) that uses a construction that, if adapted in a proper way, might yield sharp bounds on the number of points. Suppose we have a (smooth) curve \mathcal{C} of genus g over \mathbb{Q} that has a known rational point $P_0 \in \mathcal{C}(\mathbb{Q})$.

The *Jacobian variety* of \mathcal{C} is an abelian variety \mathcal{J} over \mathbb{Q} of dimension g . That means that it is a complete variety with a point $O \in \mathcal{J}(\mathbb{Q})$ together with a morphism $\mathcal{J} \times \mathcal{J} \rightarrow \mathcal{J}$ over \mathbb{Q} that defines a group operation on the points of \mathcal{J} with O as a neutral element. The points of \mathcal{J} coincide with the degree 0 divisor classes $\text{Pic}^0(\mathcal{C})$ and the fact that $\mathcal{C}(\mathbb{Q}) \neq \emptyset$ guarantees that $\mathcal{J}(\mathbb{Q}) \simeq \text{Pic}^0(\mathcal{C})(\mathbb{Q})$, i.e. the group of degree 0 divisors over \mathbb{Q} modulo linear equivalence.

Also, the map $P \mapsto [P - P_0]$ gives an injective morphism $\mathcal{C} \rightarrow \mathcal{J}$ over \mathbb{Q} . Therefore, we can consider \mathcal{C} as a subvariety of \mathcal{J} over \mathbb{Q} . As such, we have

$\mathcal{C}(\mathbb{Q}) \subset \mathcal{J}(\mathbb{Q})$. Chabauty's method applies to any curve in an abelian variety, so for the sequel it is sufficient to consider \mathcal{J} an abelian variety of dimension g over \mathbb{Q} with a curve \mathcal{C} as a subvariety over \mathbb{Q} .

The set of rational points $\mathcal{J}(\mathbb{Q})$ forms a finitely generated abelian group, the Mordell-Weil group. Therefore, there exist an $r \in \mathbb{Z}_{\geq 0}$ and a finite group T such that $\mathcal{J}(\mathbb{Q}) \simeq \mathbb{Z}^r \oplus T$. The number r is called the rank of $\mathcal{J}(\mathbb{Q})$.

Let p be a rational prime. Then $\mathcal{J}(\mathbb{Q}_p)$ is a g -dimensional p -adic analytic abelian Lie-group. Such groups are locally isomorphic to $(\mathbb{Z}_p)^g$. The topological closure of a finitely generated subgroup of rank r in such a variety will be of dimension $\leq r$. Therefore, if the Mordell-Weil rank r of $\mathcal{J}(\mathbb{Q})$ is smaller than g , then the topological closure $\overline{\mathcal{J}(\mathbb{Q})}$ of the Mordell-Weil group will be a proper analytic subvariety of $\mathcal{J}(\mathbb{Q}_p)$. Since both $\mathcal{C}(\mathbb{Q}) \subset \mathcal{J}(\mathbb{Q})$ and $\mathcal{C}(\mathbb{Q}) \subset \mathcal{C}(\mathbb{Q}_p) \subset \mathcal{J}(\mathbb{Q}_p)$, we see that

$$\mathcal{C}(\mathbb{Q}) \subset \mathcal{C}(\mathbb{Q}_p) \cap \overline{\mathcal{J}(\mathbb{Q})}.$$

The right hand side is the intersection of analytic subvarieties. If \mathcal{J} is indeed the Jacobian of \mathcal{C} , then Chabauty has proved that $\mathcal{C}(\mathbb{Q}_p)$ does not have dimension 1 intersections with proper analytic subvarieties of $\mathcal{J}(\mathbb{Q}_p)$. Therefore, the intersection is of dimension 0 and, since $\mathcal{J}(\mathbb{Q}_p)$ is compact, is finite.

This proves that $\mathcal{C}(\mathbb{Q})$ is finite and that $\#\mathcal{C}(\mathbb{Q}_p) \cap \overline{\mathcal{J}(\mathbb{Q})}$ gives an upper bound for $\#\mathcal{C}(\mathbb{Q})$. In practise, it turns out that p can often be chosen such that the bound is sharp. Furthermore, if one can find $G_1, \dots, G_m \in \mathcal{J}(\mathbb{Q})$ such that $\langle G_1, \dots, G_m \rangle = \overline{\mathcal{J}(\mathbb{Q})}$, then one can often approximate the points in $\mathcal{C}(\mathbb{Q}_p) \cap \overline{\mathcal{J}(\mathbb{Q})}$ well enough to count them.

Thus, to be able to apply this method to $\mathcal{C}_{p,t}(\mathbb{Q})$, we must estimate the Mordell-Weil ranks of the corresponding Jacobians. A theorem of Stoll (see [19] and [20]) helps.

Let l be an odd prime, let A be a non-zero, $2l$ -th power free integer prime to l and let \mathcal{C} be the smooth, complete curve with an affine model $Y^2 = X^l + A$. Let ζ_l be a primitive l th root of unity and define $q_l(A) := (A^{l-1} - 1)/l$. We assume that $-A/4^{l-1}$ is an odd, positive integer in which each prime factor occurs to an odd power.

Define

$$d_l(A) := \begin{cases} \lfloor \frac{1}{4}(l-1) \rfloor - (-1)^{\frac{1}{2}(l-1)} & \text{if } A \cdot q_l(A) \bmod l \text{ is a nonzero square in } \mathbb{F}_l, \\ \lfloor \frac{1}{4}(l-1) \rfloor & \text{otherwise.} \end{cases}$$

Theorem 3 (Stoll) *Let \mathcal{C} be the smooth, complete curve with model $Y^2 = X^l + A$ satisfying the conditions above. If the ideal class number of $\mathbb{Q}(\sqrt{l}A, \zeta_l)$ is prime to l , then the rank of the Mordell-Weil group of the Jacobian of \mathcal{C} is bounded above by $d_l(A)$.*

It should be noted that Stoll's original theorem gives a more precise statement for many more values of A , but this result suffices for us. The proof is based on a descent argument utilising the ζ_l -action on the Jacobian, described by Schaefer (see [17] and [16]).

In order to apply the theorem to our situation, we have to check that the ideal class number of $\mathbb{Q}(\sqrt{-3}, \zeta_p) = \mathbb{Q}(\zeta_{3p})$ is not divisible by p . We do this using lower bounds on discriminants together with a little bit of class field theory.

Let K be a number field. Let $\Delta(K)$ be the discriminant of the ring of integers of K . We define the *root discriminant* by $\text{rd}(K) := |\Delta(K)|^{1/[K:\mathbb{Q}]}$. If p divides the ideal class number of K , then there is a degree p unramified relative extension L of K . Such an extension has $[L : \mathbb{Q}] = p[K : \mathbb{Q}]$ and $\text{rd}(L) = \text{rd}(K)$. One can compute lower bounds on $\text{rd}(K)$, increasing in $[K : \mathbb{Q}]$ (see [15]). Thus, if $\text{rd}(K)$ is small enough, this puts a bound on the ideal class number of K .

So, if the ideal class number of $\mathbb{Q}(\zeta_{3p})$ is divisible by p , then there is a number field L with $[L : \mathbb{Q}] = 2p(p-1)$ and $\text{rd}(L) = \text{rd}\mathbb{Q}(\zeta_{3p})$. In [14] we find the following lower bounds on $\text{rd}(L)$.

p	$\text{rd}(\mathbb{Q}(\zeta_{3p}))$	$[L : \mathbb{Q}]$	lower bound on $\text{rd}(L)$
5	5.79	40	12.96
7	8.77	84	15.87
11	14.99	220	18.59
13	18.18	312	> 19.23
17	24.66	544	< 23 (> 26.48 under GRH)

So, obviously, for $p = 5, 7, 11, 13$ such L cannot exist. Therefore, Theorem 3 applies to $\mathcal{C}_{p,t}$ and we find

p	t	$d_p(A_{p,t})$	genus($\mathcal{C}_{p,t}$)	p	t	$d_p(A_{p,t})$	genus($\mathcal{C}_{p,t}$)
5	-1	1	2	11	-1	3	5
5	0	1	2	11	0	3	5
5	1	0	2	11	1	3	5
7	-1	1	3	13	-1	3	6
7	0	1	3	13	0	2	6
7	1	1	3	13	1	2	6

So, we have proved that for $p = 5, 7, 11, 13$, Chabauty methods may be applied to bound the number of primitive solutions to $x^3 + y^3 = z^p$.

Amusingly, the root discriminant argument breaks down for $p = 17$ (although not under assumption of the generalised Riemann hypothesis). This is blissfully irrelevant for the equation $x^3 + y^3 = z^p$, since Kraus's result applies.

3 The Equation $x^3 + y^3 = z^4$

The construction of the curves $\mathcal{C}_{p,t}$ does not depend on $p \geq 5$. In constructing the model $Y^2 = X^p + A_{p,t}$ we did use that p is odd, though, so we cannot use those models for $p = 4$. As it turns out, $\mathcal{C}_{4,-1}$, $\mathcal{C}_{4,0}$ and $\mathcal{C}_{4,1}$ are genus 1 curves with a rational point. Therefore, they are elliptic curves over \mathbb{Q} . Unfortunately, $\mathcal{C}_{4,-1}$ and $\mathcal{C}_{4,0}$ have Mordell-Weil groups of rank 1, so they have infinitely many rational points. This is useless if we want to bound the number of primitive solutions to $x^3 + y^3 = z^4$. For this case, we find other parametrising curves,

of higher genus. We use that fourth powers are squares. Thus, a solution to $x^3 + y^3 = z^4$ is also a solution of $x^3 + y^3 = v^2$, with $v = z^2$. We use Zagier's result to describe those solutions.

Lemma 1 (Zagier). *Let $x, y, z \in \mathbb{Z}$ be coprime integers satisfying $x^3 + y^3 = z^2$. Then there exist $s, t \in \mathbb{Z}_{\{2,3\}}$ with $(s, t) \neq (0, 0) \pmod{p}$ for any prime $p \nmid 6$ such that*

$$\begin{cases} x \text{ or } y = s^4 + 6s^2t^2 - 3t^4 \\ y \text{ or } x = -s^4 + 6s^2t^2 + 3t^4 \\ z = 6st(s^4 + 3t^4) \end{cases} \quad \text{or} \quad \begin{cases} x \text{ or } y = \frac{1}{4}(s^4 + 6s^2t^2 - 3t^4) \\ y \text{ or } x = \frac{1}{4}(-s^4 + 6s^2t^2 + 3t^4) \\ z = \frac{3}{4}st(s^4 + 3t^4) \end{cases}$$

$$\text{or} \quad \begin{cases} x \text{ or } y = s(s^3 + 8t^3) \\ y \text{ or } x = 4t(t^3 - s^3) \\ \pm z = s^6 - 20s^3t^3 - 8t^6 \end{cases}$$

See [2]. A detailed account of how to arrive at the parametrisations for $x^2 + y^4 = z^3$ can be found in [3, Section 3.2]. The proof of Lemma 1 proceeds similarly.

For a solution of $x^3 + y^3 = v^2$ to be a solution of $x^3 + y^3 = z^4$ as well, we see that the z in Lemma 1 should be a square. So, by change of variables, we see that a primitive solution to $x^3 + y^3 = z^4$ gives rise to a rational point on one of the genus 2 curves

$$\begin{aligned} \mathcal{C}_1 : Y^2 &= 6X(X^4 + 3) \\ \mathcal{C}_2 : Y^2 &= 3X(X^4 + 3) \\ \mathcal{C}_3 : Y^2 &= X^6 - 20X^3 - 8 \\ \mathcal{C}_4 : -Y^2 &= X^6 - 20X^3 - 8. \end{aligned}$$

So we can determine the primitive solutions of $x^3 + y^3 = z^4$ by finding those rational points and tracing them back to solutions of $x^3 + y^3 = z^4$. We hope that Chabauty methods apply to these curves and, as it turns out, things work out very well.

Lemma 2. *The Mordell-Weil groups of the Jacobians of the curves \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 and \mathcal{C}_4 are finite.*

Proof: This can be showed by determining the size of $\text{Jac}(\mathcal{C}_i)(\mathbb{Q})/2\text{Jac}(\mathcal{C}_i)(\mathbb{Q})$ by means of a 2-descent as described in, for instance [5]. This is quite a complicated procedure to carry out by hand but, fortunately, completely automated (see [21]). We will not bother the reader with boring details. \square

Therefore, it is sufficient to determine the torsion part of the Mordell-Weil groups. For that, we use a trick that often works and is computationally very easy. Consider a curve \mathcal{C} of genus g over \mathbb{Q}_p , given by a smooth projective model over \mathbb{Z}_p . Suppose that the reduction of that model, $\mathcal{C} \pmod{p}$, is again a smooth curve over \mathbb{F}_p . Let \mathcal{J} be the Jacobian of \mathcal{C} . Then reduction modulo p induces a splitting of $\mathcal{J}(\mathbb{Q}_p)$

$$0 \rightarrow \mathcal{J}^{(1)}(\mathbb{Q}_p) \rightarrow \mathcal{J}(\mathbb{Q}_p) \rightarrow (\mathcal{J} \pmod{p})(\mathbb{F}_p) \rightarrow 0.$$

The kernel of reduction, denoted by $\mathcal{J}^{(1)}(\mathbb{Q}_p)$, is a \mathbb{Z}_p -module. Consequently, it has only p -power torsion. By a more involved argument ([18, Theorem IV.6.4]

or [5, Theorem 7.4.1]), it follows that the kernel of reduction is actually free of torsion. We will only use that all torsion of $\mathcal{J}(\mathbb{Q}_p)$ prime to p maps injectively to $(\mathcal{J} \bmod p)(\mathbb{F}_p)$. Since $\mathcal{J}(\mathbb{Q})$ injects in $\mathcal{J}(\mathbb{Q}_p)$ for all p , we have for any pair of primes p, q of good reduction that

$$\#\mathcal{J}(\mathbb{Q})^{\text{tor}} \mid \gcd(p^r \#(\mathcal{J} \bmod p)(\mathbb{F}_p), q^s \#(\mathcal{J} \bmod q)(\mathbb{F}_q)) \text{ for some } r, s.$$

If \mathcal{C} is given by $Y^2 = F(X)$, where F is some squarefree polynomial over \mathbb{Z}_p , where p is an odd prime not dividing the discriminant of F , then \mathcal{C} can be given by a smooth model with smooth reduction, so the same principle holds (see [5, Theorem 7.4.1] for an even stronger result).

In order to count points on Jacobians, we first have to represent them. We briefly review some standard results that can be found in [5]. Let \mathcal{C} be a genus 2 curve over a field K and let \mathcal{J} be its Jacobian. A point in $\mathcal{J}(K)$ can be represented by a divisor of \mathcal{C} over K - i.e., a formal linear combination of points on \mathcal{C} . Suppose we either have one Weierstrass point $\infty \in \mathcal{C}(K)$ or two points ∞^+, ∞^- rational over K or conjugate quadratic over K that are interchanged by the hyperelliptic involution on \mathcal{C} . Then we can represent each divisor class by $[P+Q-2\infty]$ or $[P+Q-\infty^+-\infty^-]$, where $P, Q \in \mathcal{C}(K)$ or quadratic conjugate over K . This representation is even unique (apart from interchanging P and Q) for all divisor classes apart from the trivial one. The trivial divisor class, the neutral element of $\mathcal{J}(K)$, is represented by any divisor that counts the zeros of a function on \mathcal{C} with multiplicity, where poles are counted as zeros with negative multiplicity.

Lemma 3. *Let \mathcal{C} be a genus 2 curve over a finite field \mathbb{F}_q and let \mathcal{J} be its Jacobian. Then*

$$\mathcal{J}(\mathbb{F}_q) = \frac{1}{2}(\#\mathcal{C}(\mathbb{F}_q))^2 + \frac{1}{2}\#\mathcal{C}(\mathbb{F}_{q^2}) - q$$

Proof: Some simple combinatorics using the fact that divisor classes are either represented by a pair of rational points or a pair of quadratic conjugate points prove this fact. Alternatively, evaluate the characteristic polynomial of the Frobenius endomorphism at 1. See [5, Section 8.2]. \square

For $i = 1, 2, 3, 4$, we write \mathcal{J}_i for the Jacobian of \mathcal{C}_i .

Lemma 4. $\mathcal{J}_1(\mathbb{Q}) = \mathcal{J}_2(\mathbb{Q}) = \{0, [(0, 0) - \infty]\}$, so $\mathcal{C}_1(\mathbb{Q}) = \mathcal{C}_2(\mathbb{Q}) = \{(0, 0), \infty\}$.

Proof: The two divisor classes given, are clearly defined over \mathbb{Q} . We find using Lemma 3 that $\#(\mathcal{J} \bmod 5)(\mathbb{F}_5) = 26$ and $\#(\mathcal{J} \bmod 7)(\mathbb{F}_7) = 64$ for both Jacobians. Since we have already seen that the Mordell-Weil groups consist solely torsion (Lemma 2), we see that $\#\mathcal{J}(\mathbb{Q}) \mid \gcd(5^r \cdot 26, 7^s \cdot 64) = 2$, which concludes the proof. \square

Lemma 5. $\mathcal{J}_4(\mathbb{Q}) := \{0, [(1 + \sqrt{3}, 0) + (1 - \sqrt{3}, 0) - \infty^+ - \infty^-]\}$, so $\mathcal{C}_4(\mathbb{Q}) = \emptyset$.

Proof: We find $\#(\mathcal{J}_4 \bmod 5)(\mathbb{F}_5) = 36$ and $\#(\mathcal{J}_4 \bmod 19)(\mathbb{F}_{19}) = 484$. Therefore, $\#\mathcal{J}_4(\mathbb{Q}) \mid 4$. Put $F_1 = X^2 - 2X - 2$, $F_2 = X^4 + 2X^3 + 6X^2 - 4X + 4$. Then

$X^6 - 20X^3 - 8 = F_1F_2$. Furthermore, the extension of \mathbb{Q} generated by a root of F_2 is $\mathbb{Q}(\sqrt{3}, \sqrt{-1})$. A localisation of this field at a prime not above $p = 2, 3$ is at most a quadratic extension of \mathbb{Q}_p . Therefore, we have x_1, x_2 , roots of F_2 , that are either rational or quadratic conjugate over \mathbb{Q}_p . In reduction, we have $[(\bar{x}_1, 0) + (\bar{x}_2, 0) - \infty^+ - \infty^-] \in (\mathcal{J}_4 \bmod p)(\mathbb{F}_p)$. Note that two times this divisor is the divisor of the function $(X - x_1)(X - x_2)$, so it represents a 2-torsion point on $(\mathcal{J}_4 \bmod p)(\mathbb{F}_p)$. So, we see that $\mathcal{J}(\mathbb{Q}) \subset \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

Using $-[(x_1, y_1) + (x_2, y_2) - \infty^+ - \infty^-] = [(x_1, -y_1) + (x_2, -y_2) - \infty^+ - \infty^-]$, we see that a divisor represents a point of order 2 *only* if $y_1 = y_2 = 0$. As we have already seen, there is only one rational divisor class with that property. \square

Note. The above proof shows that it is impossible to bound the torsion of $\mathcal{J}_4(\mathbb{Q})$ purely by local data at primes of good reduction, since \mathcal{J}_4 has extra 2-torsion at all good primes. An alternative proof would be the following. Upon closer inspection, $(\mathcal{J}_4 \bmod 5)(\mathbb{F}_5)$ has the structure $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$. Therefore, $\mathcal{J}_4(\mathbb{Q})$ is either $\mathbb{Z}/2\mathbb{Z}$ or $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. From the data used to prove Lemma 2, we know that $\mathcal{J}_4(\mathbb{Q})/2\mathcal{J}_4(\mathbb{Q}) \simeq \mathbb{Z}/2\mathbb{Z}$.

Incidentally, this curve is a nice example of the fact that local means are not always sufficient to determine torsion. This adds importance to the height theory on genus 2 curves, which does give an effective procedure, as is described in [5, page 82] and [22]). For determining $\mathcal{C}_4(\mathbb{Q})$, however, it is entirely unnecessary to consider $\mathcal{J}_4(\mathbb{Q})$. It is straightforward to check that $\mathcal{C}_4(\mathbb{Q}_2) = \mathcal{C}_4(\mathbb{Q}_3) = \emptyset$.

Lemma 6. $\mathcal{J}_3(\mathbb{Q}) = \langle [(1 + \sqrt{3}, 0) + (1 - \sqrt{3}, 0) - \infty^+ - \infty^-], [\infty^+ - \infty^-] \rangle$ and $\#\mathcal{J}_3(\mathbb{Q}) = 6$. Furthermore, $\mathcal{C}_3(\mathbb{Q}) = \{\infty^+, \infty^-\}$.

Proof. Note that the divisor of the rational function $y - x^3 + 10$ is $\pm(3\infty^+ - 3\infty^-)$. Therefore, $3[\infty^+ - \infty^-] = 0$.

This proves that the points mentioned in the lemma generate a group of 6 elements. It remains to show that there are no other points. Upon inspection, we find $\#(\mathcal{J}_3 \bmod 5)(\mathbb{F}_5) = \#(\mathcal{J}_3 \bmod 7)(\mathbb{F}_7) = 36$, and the group structures are $\mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/6\mathbb{Z}$. Using the same argument as in Lemma 5, we find that $\mathcal{J}_3(\mathbb{Q})$ has only one point of order 2. This means that either $\mathcal{J}_3(\mathbb{Q}) \simeq \mathbb{Z}/6\mathbb{Z}$ or $\mathcal{J}_3(\mathbb{Q}) \simeq \mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$.

First we prove that $\mathcal{C}(\mathbb{Q})$ has no affine point. Suppose $P = (x, y) \notin \{\infty^+, \infty^-\}$ with $y^2 = x^6 - 20x^3 - 8$. If $P \in \mathcal{C}_3(\mathbb{Q})$, then $[P - \infty^+] \in \mathcal{J}_3(\mathbb{Q})$, so $0 = 6[P - \infty^+] = [6P - 3\infty^+ - 3\infty^-] - 3[\infty^+ - \infty^-] = 3[2P - \infty^+ - \infty^-]$. We construct a function g that has (at least) a quadruple zero in P and has triple poles in ∞^+ and ∞^- . As it turns out, this fixes the location of the other two zeros. Therefore, if $6[P - \infty^+] = 0$, then g has a zero of order 6 in P .

Put $F(X) = X^6 - 20X^3 - 8$ and let $g(X, Y) = Y - (g_3X^3 + g_2X^2 + g_1X + g_0)$ be a function on \mathcal{C}_3 with a quadruple zero at P and a triple pole at both ∞^+ and ∞^- . Since $F(X)$ has no rational roots, without loss of generality, $y^2 = F(x) \neq 0$, so $t = X - x$ is a uniformising coordinate at P . We compute a power series expansion of g in t at P .

$$\theta_P := \theta_P^{(0)} + \theta_P^{(1)}t + \dots + \theta_P^{(5)}t^5 + O(t^6) = g(x + t, \sqrt{F(x + t)}).$$

Since $g(P) = 0$, we have $g_0 = y - g_3x^3 - g_2x^2 - g_1x$. We solve g_1, g_2, g_3 from the equations $\theta_P^{(1)} = \theta_P^{(2)} = \theta_P^{(3)} = 0$, which must hold for P to be a quadruple zero of g . We find

$$\begin{aligned}\theta_P^{(4)} &= 1620 \frac{x^2(x-1)^2(x+2)^2(x^2-2x+4)^2(x^2+x+1)^2}{F(x)^3}, \\ \theta_P^{(5)} &= 648x(x-1)(x+2)(x^2-2x+4)(x^2+x+1)(x^3+2) \\ &\quad (4x^9-111x^6-168x^3+32)/F(x)^4.\end{aligned}$$

Thus, we see that the only P with $6[P-\infty^+] = 0$ and $X(P) \in \mathbb{Q}$ are $(0, 2\sqrt{-2})$, $(1, 3\sqrt{-3})$ and $(-2, 6\sqrt{6})$, and none of these points has $Y(P) \in \mathbb{Q}$. Therefore, $\mathcal{C}_3(\mathbb{Q}) = \{\infty^+, \infty^-\}$.

It follows that any 3-torsion point in $\mathcal{J}_3(\mathbb{Q})$ other than $\pm[\infty^+ - \infty^-]$, is of the form $[(x_1, y_1) + (x_2, y_2) - \infty^+ - \infty^-]$, where (x_1, y_1) and (x_2, y_2) are quadratic conjugate points over \mathbb{Q} and $y_1 \neq 0$. We use the same construction to show that such points do not exist. We determine g_0, \dots, g_3 such that the function $Y - g_3X^3 - g_2X^2 - g_1X - g_0$ has double zeros in (x_1, y_1) and (x_2, y_2) . Then we determine for which point the function has in fact zeros of order 3 in those points. One can do this by computing the conditions $x_1 + x_2$ and x_1x_2 . The computer algebra involved in this computation, is too bulky to display here, but completely straightforward. There turns out to be no other 3-torsion. \square

Note. The proof of Lemma 6 exhibits that $6[(0, 2\sqrt{-2}) - \infty^+] = 6[(1, 3\sqrt{-3}) - \infty^+] = 6[(-2, 6\sqrt{6}) - \infty^+] = 0$. For any prime $p \neq 2$, this yields an extra 6-torsion point over \mathbb{Q}_p . This gives another example of a curve for which the torsion of the Jacobian cannot be determined solely by information at primes of good reduction.

Proof of Theorem 1: Lemmas 4 through 6 give us the rational points on the curves \mathcal{C}_1 through \mathcal{C}_4 . These points all correspond to trivial solutions of $x^3 + y^3 = z^4$. Thus, by construction of the curves, the only primitive solutions are trivial. \square

4 The Equation $x^3 + y^3 = z^5$

As was shown in Section 2, it is sufficient to determine the rational points on the curves

$$\begin{aligned}\mathcal{C}_1 : Y^2 &= X^5 - 559872 \\ \mathcal{C}_2 : Y^2 &= X^5 - 62208 \\ \mathcal{C}_3 : Y^2 &= X^5 - 6912\end{aligned}$$

in order to find all primitive solutions to $x^3 + y^3 = z^5$. Let \mathcal{J}_i be the Jacobian of \mathcal{C}_i . As a result of Theorem 3, we have seen that $\text{rk}(\mathcal{J}_1(\mathbb{Q})) \leq 1$, $\text{rk}(\mathcal{J}_2(\mathbb{Q})) \leq 1$ and that $\text{rk}(\mathcal{J}_3(\mathbb{Q})) = 0$. The procedure of determining the rational points on curves of genus 2 is, provided that generators of the appropriate Mordell-Weil groups can be found, almost a standard one, although not guaranteed to be successful. Instead of providing a lot of hard to check numerical data to make

up the proof, we will describe a session with the available software to give an idea how one can perform these computations in practice nowadays.

Lemma 7. $\mathcal{C}_1(\mathbb{Q}) = \{\infty\}$

We will prove this lemma using Chabauty techniques as described in [5], [10] and [4]. In these articles, the curve is embedded in the Jacobian using the map $P \mapsto [2P - \infty^+ - \infty^-]$ or $[2P - 2\infty]$. We describe a session with the software mentioned in [4].

```
> read'divcalc.mpl';
> initcurve(x^5+A);
'current curve is ', y^2 = x^5+A
> s_nG:=locexp([n*L[1],n*L[2]]):
> k_nG:=loccoord2kummer(s_nG):
> theta_nG:=factor(series(kummer2theta(k_nG),n,11));
theta_nG := series((4*L[2]^5*(A*L[2]^5+L[1]^5))*n^10+O(n^12),n,12)
> coeff_theta:=factor(coeff(theta_nG,n,10));
coeff_theta := 4*L[2]^5*(A*L[2]^5+L[1]^5)
```

Let $G \in \mathcal{J}(\mathbb{Q})$ be a point in the kernel of reduction mod p of the Jacobian of a curve $Y^2 = X^5 + A$. The two quantities L_1 and L_2 can be computed from G and if the quantity `coeff_theta` does not vanish mod p^{12} , then nG is not of the form $[2P - 2\infty]$ for any $n \neq 0$. Below, G will be a point in $\mathcal{J}(\mathbb{Q})$. It is a generator of the Mordell-Weil group, but we will only need and check that it generates a group of finite index not divisible by certain primes.

```
> alias(alpha=RootOf(x^2+3));
I, alpha
> F:=x^5-559872;
F := x^5-559872
> P:=[12*alpha,1296+864*alpha];
P := [12*alpha, 1296+864*alpha]
> G:=[P,conj(P)];
G := [[12*alpha, 1296+864*alpha], [-12*alpha, 1296-864*alpha]]
> initcurve(F);
'current curve is ', y^2 = x^5-559872
> njac(11,F);
131
> njac(19,F);
400
```

The last two instructions count the number of points on the Jacobian in reduction. The values are coprime, so there is no rational torsion.

```
> alpham19:=(Roots(op(alpha))mod 19);
alpham19 := [[15, 1], [4, 1]]
> Gred19:=subs(alpha=alpham19[1][1],G)mod 19:
> tbl19:=maketablep(19,F mod 19,[[]],Gred19):
> rowdim(tbl19);
```

```

20
> select(i->tbl19[i+1,2][1][1]=tbl19[i+1,2][2][1],
   [i$i=1..rowdim(tbl19)-1]);
[8, 12]
> G7red29:=adp(29,mlp(29,3,G),mlp(29,4,G));
G7red29 := [[21, 27], [25, 15]]
> tbl29:=maketablep(29,F mod 29,[[]],G7red29):
> rowdim(tbl29);
30
> select(i->tbl[i+1,2][1][1]=tbl[i+1,2][2][1],
   [i$i=1..rowdim(tbl29)-1]);
[]
> alpham7:=(Roots(op(alpha))mod 7);
alpham7 := [[2, 1], [5, 1]]
> Gred7:=subs(alpha=alpham7[1][1],G)mod 7:
> njac(7,F);
50
> mlp(7,10,Gred7);mlp(7,25,Gred7);
[[0, 4], [0, 4]]
[[3, 0], [infinity, 0]]

```

This shows that $20G \in \mathcal{J}^{(1)}(\mathbb{Q}_{19})$ and that any divisor $nG = [2P - 2\infty]$ has $n = 20m$ or $n = 8 + 20m$ or $n = 12 + 20m$ for some $m \in \mathbb{Z}$. However, the reduction data at 29 shows that any such point has $n = 30m$ for some $m \in \mathbb{Z}$, so this rules out the last two options. We see that G generates the full group $(\mathcal{J} \bmod 7)(\mathbb{F}_5)$ of order 50. Therefore, the group generated by G has index prime to 10 in the Mordell-Weil group.

```

> Gml20:=ml(20,G):
> sGml20:=div2loccoord(Gml20):
> lGml20:=loclog(sGml20)mod 19^3;
lGml20 := [4636, 2660]
> subs(A=-559872,L[1]=lGml20[1]/19,L[2]=lGml20[2]/19,
   coeff_theta)mod 19;
14

```

Here, we compute $20G$ and its 19-adic logarithm. The fact that $[4636, 2660] \neq [0, 0] \bmod 19^2$, proves that $19 \nmid [\mathcal{J}(\mathbb{Q}) : \langle G \rangle]$. Furthermore, we check that indeed the coefficient of n^{10} in θ does not vanish $\bmod 19^{11}$. This proves that $0G$ is the only multiple of $20G$ of the form $[2P - 2\infty]$. See [4] for details on how this power series argument works.

Lemma 8. $\mathcal{C}_2(\mathbb{Q}) = \{\infty\}$.

Although in principle the same procedure applies to this curve as well, we will use another method, described in [3]. An advantage of this method is that it might still work if $\text{rk}(\mathcal{J}(\mathbb{Q})) > 1$. Put $\alpha^5 = 2$. We use that if $x, y \in \mathbb{Q}$ with $y^2 = x^5 - 62208$, then there is a δ in some finite set and $y_1, y_2 \in K = \mathbb{Q}(\alpha)$ such that

$$\begin{aligned} x - 6\alpha^3 &= \delta y_1^2 \\ x^4 + 6\alpha^3x^3 + 72\alpha x + 432\alpha^4x + 5184\alpha^2 &= \delta y_2^2 \end{aligned}$$

We can therefore suffice in finding the K -rational points on genus 1 curves with an X -coordinate in \mathbb{Q} . It turns out we can suffice in proving

Lemma 9. *The K -rational points on the genus 1 curve $Y^2 = X^4 + 6\alpha^3X^3 + 72\alpha X + 432\alpha^4X + 5184\alpha^2$ with rational X -coordinate have $X \in \{\infty, 0, -12\}$.*

Lemma 10. *The K -rational points on the genus 1 curve $(1 + 2\alpha - 2\alpha^2 + 2\alpha^3 - 2\alpha^4)Y^2 = X^4 + 6\alpha^3X^3 + 72\alpha X + 432\alpha^4X + 5184\alpha^2$ with rational X -coordinate have $X \in \{12\}$.*

Note that all finite X -coordinates found satisfy $X^5 < 62208$, so they do not correspond to rational point on \mathcal{C}_2 . Therefore, $\mathcal{C}_2(\mathbb{Q}) = \{\infty\}$. More details on how to do this can be found in [3]. Here, we describe a session with a package that does these computations for you.

```
kash> Read("ell.g");
ell package loaded.
kash> pol:=x^5-62208;
x^5 - 62208
kash> O:=OrderMaximal(x^5-2);
Generating polynomial: x^5 - 2
Discriminant: 50000
kash> OrderClassGroup(O);
[ 1, [ 1 ] ]
kash> alpha:=XOrderPrimElt(O);
[0, 1, 0, 0, 0]
kash> theta:=PolyRoots(pol+RingZero(O))[1];
[0, 0, 0, 6, 0]
kash> Qpol:=x-theta;
x + [0, 0, 0, -6, 0]
kash> Rpol:=pol/Qpol;
x^4+[0,0,0,6,0]*x^3+[0,72,0,0,0]*x^2+[0,0,0,0,432]*x+
[0,0,5184,0,0]
kash> deltas:=FilterTwists(Qpol,Rpol);
[ 1, [1, 2, -2, 2, -2] ]
```

The last result shows that indeed we only need consider two twists of the genus 1 curve (`FilterTwists` checks which twists have points with the desired property locally).

```
kash> ec:=Quar(x^3-5*x^2+Elt(0,5)*x);
kash> EllAddHint(ec,1);EllAddHint(ec,1-2*alpha+alpha^3);
kash> EllGensMod2(ec);
Finding generators of 2-isogeny selmer group on curve...
```

:

```
Found global basis using hints.
[(0:0:1),([1,-2,0,1,0]:[3,0,-2,-2,1]:1),(1:1:1)]
kash> EllGenInit([EllXtoPnt(ec,1),
>     EllXtoPnt(ec,1-2*alpha+alpha^3),EllXtoPnt(ec,0)],2);
```

The function `Quar` initialises an elliptic curve in Weierstrass form from a model of the form $Y^2 = F(X)$. Since in this call, F is cubic, transforming to Weierstrass form is trivial. The next commands give hints to the system about where to look for X -coordinates of Mordell-Weil generators. The command `EllGensMod2` tries to find generators of $E(K)/2E(K)$. First it bounds the rank using a 2-descent or a 2-isogeny descent (see [18]), depending on the curve. Then it makes a feeble attempt at finding generators. This is where giving hints helps tremendously. We register the found points. We will only prove that certain primes do not divide the index of the generated group in the full Mordell-Weil group. Note that we already know that the index is odd.

```
kash> cov:=QuarCov(Rpol,ec);;
kash> p151:=PlaceSupport(151*0);;
kash> List(p151,p->EllGrpIndex(ec mod p));
[ 2, 2, 2, 1, 1 ]
```

This shows that the images of the group we have determined in the reductions at the several places over 151 (this is the smallest completely split prime. It is not necessary to use a split prime, but it does reduce the amount of needed computations) are at most 2. Since we know the index to be odd from the descent, we know that the image subjects on the reduction of the Mordell-Weil group. Finally, we check which points P have a rational X -coordinate on the original quartic model. Note that for such a point, the values of $X(P)$ under all 5 embeddings $K \rightarrow \mathbb{Q}_{151}$ should agree. This gives 4 equations. Since the Mordell-Weil rank is 2, there are essentially only two variables, so we expect only finitely many solutions. See [3] for details.

```
kash> EllCovChab(cov,[0,[-12,1],[1,0]],p151);
Result of FibStrict:[151,[[139,1],[0,1],[1,0]]]
Computing Theta^G for G=( 4: 2: 1 )...
G is only point in fiber if matrix has maximal rank mod 151
[ 9 60]
[132 113]
[ 1 115]
[ 73 101]

Computing Theta^G for G=([4,-2,-3,0,2]:[2,-9,-7,4,7]:1)...
G is only point in fiber if matrix has maximal rank mod 151
[ 44 55]
[145 147]
[140  2]
[ 12 17]

Computing Theta^G for G=( 0: 1: 0 )...
Point maps to infinity. Taking 1/phi
G is only point in fiber if matrix has maximal rank mod 151
[ 37 56]
[106 79]
[100 66]
```

```
[ 48 28]
```

```
[ [ 0, 1 ], [ -12, 1 ], [ 1, 0 ] ]
kash>
```

For the other elliptic curve, we proceed similarly.

```
kash> ec:=Quar((x^3-5*x^2+Elt(0,5)*x)/deltas[2]);
Elliptic curve [1,2,-2,2,-2]*y^2=x^3-5*x^2+5*x over order
generated by x^5 - 2
kash> cov:=QuarCov(Rpol/deltas[2],12,ec);;
kash> p3:=PlaceInit((-1+alpha-alpha^2+alpha^3-alpha^4)*0);
place [ 3, [1, 2, 1, 2, 1] ] above 3
kash> EllAddHint(ec,Elt(0,[8,6,5,4,3]));
kash> EllAddHint(Ell2Iso(ec),Elt(0,[109,128,112,80,52])/9);
kash> EllGensMod2(ec);
Finding generators of 2-isogeny selmer group on curve...
Computing 2-isogeny selmer rank.
```

```
:
```

```
Found global basis using hints.
[(0:0:1),([8,6,5,4,3]:[76,67,58,50,43]:1),
 ([14308,8384,9136,4952,4324]/2601:[-4461238,-3728612,-3260650,
 -2970062,-2572342]/132651:1)]
kash> EllGenInit([EllXtoPnt(ec,Elt(0,[8,6,5,4,3])),
 >           EllXtoPnt(ec,Elt(0,[15, -20, 5, -10, 10])/ 12),
 >           EllXtoPnt(ec,0)],2);
kash> List(p151,p->EllGrpIndex(ec mod p));
[ 1, 2, 2, 1, 8 ]
kash> EllCovChab(cov,[12],p151);
Result of FibStrict:[ 151, [ [ 12, 1 ] ] ]
Computing Theta^G for G=( 0: 1: 0 )...
G is only point in fiber if matrix has maximal rank mod 151
[148 37]
[ 31 98]
[113 95]
[ 41 76]

[ [ 12, 1 ] ]
kash>
```

Lemma 11. $\mathcal{C}_3 = \{\infty\}$.

Proof: Since we already know that the Jacobian has rank 0, we only need to determine the torsion. The fact that $(\mathcal{J}_3 \bmod 7)(\mathbb{F}_7) = 43$ and $(\mathcal{J}_3 \bmod 11)(\mathbb{F}_{11}) = 375$ shows that there is none. \square

5 Availability of Programs

Two implementations of the descent procedure on genus 2 curves as described in [5] are available, both written by Stoll. One is based on the public domain packages PARI/GP ([1]) and CLISP, the other is a package for MAGMA. The latter also has routines for applying Chabauty methods on Jacobians of genus 2 curves. The PARI/GP based program is available in binary form for Linux/i386 systems from

<http://www.math.uni-duesseldorf.de/~stoll/genus2/>

and the MAGMA package can be obtained from

[http://www.math.uni-duesseldorf.de/~stoll/programs/HC/.](http://www.math.uni-duesseldorf.de/~stoll/programs/HC/)

Routines for doing computations on Jacobians of genus 2 curves are available from several locations. There are some routines referred to in [5] at <ftp://ftp.liv.ac.uk/pub/genus2> for Maple V. The author has also made available some routines for the commercial computer algebra package Maple V for doing computations. An example session is described in Section 4. In the same location, there are also some rudimentary routines for doing 2-descents, written for KASH ([7]).

An elliptic curve package for elliptic curves over arbitrary number fields based on KASH is also available. It can do 2-descent and 2-isogeny descent on elliptic curves over number fields, also with even class number. Furthermore, as demonstrated, there are facilities for Chabauty-arguments.

The computations needed to determine the 3-torsion in Section 3, can be found in `tor334.mpl`. The proof of Lemma 7 can be found in `prf335.mpl`, but the Maple program in `divcalc.sh`, available from the same location, is also necessary. The elliptic curve Chabauty method can be found in `prf335.g`, together with the KASH package `ell.sh` which it is based on.

Electronic locations have a very temporary nature. All these files are presently located at <http://www.math.uu.nl/people/Bruin/>, but this will probably not be permanent. The author will attempt to have a home page somewhere with links to the relevant files.

Acknowledgements

I wish to thank Frits Beukers for the stimulating discussions and pointing out a much more efficient parametrisation of the solutions of $x^3 + y^3 = z^p$. The rank computation programs and the quick and accurate replies of Michael Stoll have also been a great help. The remarks of Alain Kraus have helped to put the work in perspective. Furthermore, I would like to thank Utrecht University for supplying the facilities and stimulating scientific environment that made this paper possible.

References

1. C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. PARI-GP. Available from <ftp://megrez.math.u-bordeaux.fr/pub/pari>.
2. Frits Beukers. The Diophantine equation $Ax^p + By^q = Cz^r$. *Duke Math. J.*, 91(1):61–88, 1998.
3. Nils Bruin. *Chabauty Methods and Covering Techniques applied to Generalised Fermat Equations*. PhD thesis, Universiteit Leiden, 1999.
4. Nils Bruin. The diophantine equations $x^2 \pm y^4 = \pm z^6$ and $x^2 + y^8 = z^3$. *Compositio Math.*, 118:305–321, 1999.
5. J.W.S. Cassels and E.V. Flynn. *Prolegomena to a Middlebrow Arithmetic of Curves of Genus 2*. LMS-LNS 230. Cambridge University Press, Cambridge, 1996.
6. Claude Chabauty. Sur les points rationnels des variétés algébriques dont l’irrégularité est supérieure à la dimension. *C. R. Acad. Sci. Paris*, 212:1022–1024, 1941.
7. M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, and K. Wildanger. KANT V4. *J. Symbolic Comput.*, 24(3-4):267–283, 1997. Available from <ftp://ftp.math.tu-berlin.de/pub/algebra/Kant/Kash>.
8. Henri Darmon and Andrew Granville. On the equations $z^m = F(x, y)$ and $Ax^p + By^q = Cz^r$. *Bull. London Math. Soc.*, 27(6):513–543, 1995.
9. Henri Darmon and Loïc Merel. Winding quotients and some variants of Fermat’s last theorem. *J. Reine Angew. Math.*, 490:81–100, 1997.
10. E.V. Flynn. A flexible method for applying chabauty’s theorem. *Compositio Mathematica*, 105:79–94, 1997.
11. Alain Kraus. private communication.
12. Alain Kraus. Sur l’équation $a^3 + b^3 = c^p$. *Experiment. Math.*, 7(1):1–13, 1998.
13. R. Daniel Mauldin. A generalization of Fermat’s last theorem: the Beal conjecture and prize problem. *Notices Amer. Math. Soc.*, 44(11):1436–1437, 1997.
14. A. M. Odlyzko. Tables of discriminant bounds. available at, 1976, <http://www.research.att.com/~amo/unpublished/index.html>.
15. A. M. Odlyzko. Bounds for discriminants and related estimates for class numbers, regulators and zeros of zeta functions: a survey of recent results. *Sém. Théor. Nombres Bordeaux (2)*, 2(1):119–141, 1990.
16. Bjorn Poonen and Edward F. Schaefer. Explicit descent for jacobians of cyclic covers of the projective line. *J. reine angew. Math.*, 488:141–188, 1997.
17. Edward F. Schaefer. Computing a Selmer group of a Jacobian using functions on the curve. *Math. Ann.*, 310(3):447–471, 1998.
18. Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. GTM 106. Springer-Verlag, 1986.
19. Michael Stoll. On the arithmetic of the curves $y^2 = x^l + A$ and their Jacobians. *J. Reine Angew. Math.*, 501:171–189, 1998.
20. Michael Stoll. On the arithmetic of the curves $y^2 = x^l + A$, II. available from <http://www.math.uiuc.edu/Algebraic-Number-Theory>, 1998.
21. Michael Stoll. Implementing 2-descent for jacobians of hyperelliptic curves. available from <http://www.math.uiuc.edu/Algebraic-Number-Theory>, 1999.
22. Michael Stoll. On the height constant for curves of genus two. *Acta Arith.*, 90(2):183–201, 1999.
23. R. Tijdeman. Diophantine equations and Diophantine approximations. In *Number theory and applications (Banff, AB, 1988)*, pages 215–243. Kluwer Acad. Publ., Dordrecht, 1989.

Factoring Polynomials over p -Adic Fields

David G. Cantor and Daniel M. Gordon

Center for Communications Research
4320 Westerra Court, San Diego, CA 92121
`{dgc,gordon}@ccrwest.org`

Abstract. We give an efficient algorithm for factoring polynomials over finite algebraic extensions of the p -adic numbers. This algorithm uses ideas of Chistov's random polynomial-time algorithm, and is suitable for practical implementation.

1 Introduction

Factoring polynomials over the p -adic numbers \mathbb{Q}_p is an important problem in computational number theory. One application is determining the prime ideals of a number field $\mathbb{Q}(\alpha)$, and how a given rational prime p factors into prime ideals in that field. See Cohen [10] and the references cited therein for some methods currently in use.

These algorithms, while generally good in practice, will take exponential time for some polynomials. A. L. Chistov ([7], [8], and [9]) has given an algorithm which runs in random polynomial time for all polynomials, but would be very difficult to implement efficiently. In this paper we give a random polynomial-time algorithm which works well in practice. The algorithm is non-deterministic only because all known efficient algorithms for factoring polynomials over finite fields \mathbb{F}_{p^n} ([3], [5]) are non-deterministic. Note that any polynomial-time p -adic factoring algorithm can factor polynomials over \mathbb{F}_{p^n} in polynomial time. It has been implemented in PARI, and is available on the second author's web site [13].

We will factor polynomials over a finite algebraic extension K of \mathbb{Q}_p . See Chapter 5 of [14] for properties of these extensions. Let π be a uniformizer of K . In the case when K is an unramified extension of \mathbb{Q}_p , we choose $\pi = p$. For x in the ring of integers O_K of K , \bar{x} will denote the image of x in the residue class field \bar{K} . We will fix a set of representatives $\mathcal{A} = \{0, a_1, \dots, a_{p^f-1}\} \subset O_K$ for the elements of \bar{K} . This set may be lifted to representatives for unramified extensions of K in a straightforward manner.

The valuation of an element $x \in K$ will be denoted $|x|$, and its order by $\text{ord } x$. We assume that $||$ has been normalized so that $|p| = 1/p$. There is a unique extension of the valuation $||$ on K to its algebraic closure \tilde{K} ; we assume that $||$ has been so extended.

Just as for real numbers, one cannot, in general, explicitly represent a p -adic number exactly, but only an approximation which is a rational number. Thus our algorithm will find approximations to factors of $F(X)$. Elements x of K may

be written $x = \sum_{i=-m}^{\infty} a_i \pi^i$, with $a_i \in \mathcal{A}$. In Section 8 we discuss where this series can be truncated to guarantee a correct answer.

Let $F(X)$ be a monic polynomial with coefficients in O_K which has no repeated factors. See Zippel [29, pp 294–295] for a simple method of removing repeated factors. Unlike Chistov’s algorithm, our method does not require computing in, or even constructing, ramified extensions of K . The algorithm is applied recursively, at each step either finding a new factor or terminating with an irreducible factor and certificate of its irreducibility. The certificate of irreducibility will be a generalized Eisenstein polynomial with coefficients in the maximal unramified (over K) subfield of $K(x)$, where x is a root of the irreducible factor.

The **p-adic Factor** algorithm works by looking for a polynomial $A(X)$ for which we can determine the factorization of

$$R(Y) = \text{Res}_X(F(X), Y - A(X)). \quad (1.1)$$

In Section 2 we show that a factor of $R(Y)$ lets us find a factor of $F(X)$, and a certificate of irreducibility for $R(Y)$ also applies to $F(X)$. Once such an $A(X)$ is found, we apply the information to $F(X)$ and, if necessary, recurse on remaining factors of the original polynomial.

The standard “easy” method for factoring a polynomial over the p -adics, the Newton diagram method, is given in Section 3. If the Newton diagram of the polynomial is not a straight line, then Hensel’s Lemma may be used to find a factor. If the Newton diagram is a straight line with slope k/n , where n is the degree of $F(X)$ and k is relatively prime to n , then $F(X)$ is irreducible.

Otherwise the Newton diagram method fails, and we use an extension of Hensel’s Lemma given in Section 4.1. We proceed by looking at the factorization of $F(X)$ in \tilde{K} . If the reduction $F^*(X)$ (defined in Section 3) has two relatively prime factors, then using Hensel’s Lemma we may lift these to factors over K . If $F^*(X)$ is the power of an irreducible polynomial of degree $d \geq 2$, then we may factor $F(X)$ over an unramified extension of degree d of K , leading to a factorization of $F(X)$ over K . These methods form the basis of the **Hensel Factor** routine given in Section 4.2. The only case **Hensel Factor** cannot handle is when

$$R(Y) = a_n (Y^r - b\pi^s)^m + [\text{terms above the Newton diagram}]. \quad (1.2)$$

In this case we have $\text{ord } A(x) = s/r$ for each root x of $F(X)$ in \tilde{K} , the closure of K . The **p-adic Factor** algorithm then finds a new polynomial $A(X)$ such that either **Hensel Factor** successfully factors $R(Y)$, or (1.2) still holds with either $\text{ord } A(x)$ or $\deg A(X)$ increased. Since $\deg A(X) < n$, and $\text{ord } A(x)$ is bounded by Corollary 5.8, this will terminate after a bounded number of steps.

In Section 7 we illustrate how the algorithm works on two examples. Section 8 gives a worst-case bound for the bit complexity of the algorithm

$$O(n^{8+\epsilon} \log^3 |\Delta_F| \log^2 p^k), \quad (1.3)$$

where n is the degree of $F(X)$, Δ_F is the discriminant of $F(X)$, and k is the degree of K over \mathbb{Q}_p .

Our algorithm may be extended to any local field complete with respect to a discrete rank-1 valuation, under the assumptions that the residue class field is perfect and that an algorithm for factoring polynomials defined over the residue-class field is given. For example, applying it to the field $\mathbb{F}_q((X))$ of Laurent series, it can be used to resolve singularities of plane curves. A future paper will extend the algorithm to other local fields, and include some proofs which have been omitted here due to space constraints.

We thank Stephen DiPippo and Robert Segal for many helpful discussions. John Cannon told us of developments with MAGMA's local rings and fields package, and informed us that the MAGMA group has developed a similar algorithm for factoring polynomials over \mathbb{Q}_p , which is currently being implemented.

2 Some Criteria for Factorization

In this section we give simple criteria for polynomial factorization and polynomial irreducibility. Let $\text{Res}_X(A(X), B(X))$ denote the resultant of two polynomials $A(X)$ and $B(X)$. See Lang [19] or Cassels [6] for details. Due to space constraints we omit proofs of the lemmas in this section. They follow in a straightforward way from the properties of the resultant.

Lemma 2.1. *Suppose that $F(X)$ and $A(X)$ are polynomials in the field $K[X]$ with $F(X)$ monic of degree n . Put*

$$R(Y) = \text{Res}_X(F(X), Y - A(X)). \quad (2.2)$$

Then

1. $R(Y)$ is a monic polynomial of degree n in Y and
2. the polynomial $F(X)$ divides the polynomial $R(A(X))$.

The following lemma provides a way of factoring a polynomial.

Lemma 2.3. *Suppose that $F(X)$ and $A(X)$ are polynomials in $K[X]$, with $F(X)$ monic. Put*

$$R(Y) = \text{Res}_X(F(X), Y - A(X)). \quad (2.4)$$

Suppose further that $R(Y) = R_1(Y)R_2(Y)$ is a factorization of $R(Y)$ into relatively prime, non-constant factors. Then

$$F(X) = F_1(X)F_2(X), \quad (2.5)$$

where

$$F_1(X) = \gcd(F(X), R_1(A(X))) \text{ and } F_2(X) = \gcd(F(X), R_2(A(X))), \quad (2.6)$$

is a factorization of $F(X)$ into relatively prime, non-constant factors. Furthermore,

$$\deg F_1(X) = \deg R_1(Y) \text{ and } \deg F_2(X) = \deg R_2(Y). \quad (2.7)$$

The following Lemma provides a partial converse to Lemma 2.3.

Lemma 2.8. *Suppose that $F(X)$ is a monic polynomial of degree n , that $A(X)$ is a polynomial, and that both have coefficients in the field K . If the polynomial $R(Y) = \text{Res}_X(F(X), Y - A(X))$ is irreducible over K , then $F(X)$ is also irreducible over K .*

If neither Lemma 2.3 nor Lemma 2.8 applies, we may need to go to an unramified extension field of K . The following lemma shows how irreducible factors of $F(X)$ over an extension field L of K lead to irreducible factors over K .

Lemma 2.9. *Suppose that $F(X)$ is a monic polynomial in $K[X]$ with no repeated factors of degree ≥ 1 , that L is a finite algebraic extension of K , and that $G(X)$ is a monic, irreducible, polynomial in $L[X]$ of degree ≥ 1 which divides $F(X)$. Put $H(X) = \text{Norm}_{L/K} G(X)$. Then,*

1. *$\gcd(F(X), H(X))$ is an irreducible factor of degree ≥ 1 of $F(X)$ in $K[X]$; and*
2. *if the field extension L/K is generated by the coefficients of $G(X)$, then $H(X)$ is already an irreducible factor of $F(X)$ in $K[X]$.*

3 Newton Diagrams

In this section we give our notation for Newton diagrams and some related items. For details see Artin [1], Cassels [6], or Gouvea [14, Section 6.4].

Suppose that

$$R(Y) = \sum_{i=0}^n a_i Y^i \tag{3.1}$$

is a polynomial in $K[Y]$ of (exact) degree $n \geq 1$. As usual, we associate to $R(Y)$ a finite, non-empty point set $\mathfrak{S} \subset \mathbb{R}^2$ consisting of points $(i, \text{ord } a_i) \in \mathbb{R}^2$ corresponding to each nonzero term $a_i Y^i$ of $R(Y)$.

Definition 3.2. We define, as is customary, the *Newton diagram* of $R(Y)$ to be the lower boundary of the convex hull of \mathfrak{S} .

Following Cassels [6], we use the following definition:

Definition 3.3. Suppose that $R(Y)$ is given by (3.1). We shall call $R(Y)$ *pure* if $a_0 \neq 0$, $n \geq 1$, and the Newton diagram of $R(Y)$ is a straight line.

If the Newton diagram is not pure, we may immediately factor $R(Y)$. The following is well known (see Cassels [6]), and is also a corollary of our Theorem 4.21.

Lemma 3.4. *Suppose that $R(Y) = \sum_{i=0}^k a_i Y^i$ is a polynomial of degree $k \geq 1$ and that a_0 is not zero. If the polynomial $R(Y)$ is not pure (so that its Newton diagram consists of two or more straight line-segments necessarily of different slopes), then $R(Y)$ factors into two non-constant polynomials in $K[Y]$.*

If the Newton diagram is pure, we may sometimes use its slope to show that $R(Y)$ is irreducible.

Lemma 3.5. (Generalized Eisenstein criterion) Suppose $R(Y)$ is pure, and its Newton diagram has slope k/n , where k is an integer relatively prime to n . Then $R(Y)$ is irreducible.

Proof. If y is a root of $R(y)$ in \tilde{K} , then $\text{ord } y = k/n$. Hence $K(y)/K$ is a totally ramified extension and has degree n , so $R(Y)$ is irreducible. \square

Remark 3.6. The customary form of Eisenstein's criterion is the special case when $k = -1$ (see, for example, [29]).

Now suppose that $R(Y)$ is pure and has slope $-s/r$. Because the points $(0, \text{ord } a_0)$ and $(n, \text{ord } a_n)$ are the end-points of the Newton diagram, n must be an integral multiple of r , say, $n = mr$. Put

$$\alpha_i = a_{ri}/(a_n \pi^{s(m-i)}) \quad (3.7)$$

so that $\alpha_i \in O_K$. We can then write

$$R(Y) = a_n \sum_{i=0}^m \alpha_i \pi^{s(m-i)} Y^{ri} + [\text{terms above the Newton diagram}]. \quad (3.8)$$

Here “terms above the Newton diagram” refers to those non-zero terms of $R(Y)$ whose corresponding points in the Newton set \mathfrak{S} lie strictly above the Newton diagram. These are the non-zero terms of the form $a_i Y^i$ for which $\text{ord } a_i > s(m-i)/r + \text{ord } a_n$.

Definition 3.9. Suppose $R(Y)$ as given by (3.1) is pure and suppose that the α_i are given by (3.7). Define

$$R^*(Y) = \sum_{i=0}^m \bar{\alpha}_i Y^i. \quad (3.10)$$

The polynomial $R^*(Y)$ is monic and has coefficients in \overline{K} . In the next section we will show how to factor $F(X)$ using Hensel's Lemma if we can write $R^*(Y)$ as the product of two relatively prime factors, perhaps over an extension field of K . Otherwise, we will use a reduction method extending the one used by Chistov [8].

4 Factoring with Hensel's Lemma

4.1 Hensel's Lemma

Hensel's Lemma refers to an algorithm, due to Hensel [17], which shows how to find a factorization of a polynomial $R(Y) \in K[Y]$ from an “approximate

factorization". Here we describe an extension of this algorithm. The extension is related to that of Artin [1]. The main novelty is Corollary 4.30. In the special case when the slope of the Newton diagram of $R(Y)$ is zero, it is well known. Dealing with general slopes avoids the need to go to ramified extension fields as in [8], making the algorithm much more practical.

Definition 4.1. Suppose that λ is a positive real number. If

$$A(Y) = \sum_{i=0}^k a_i Y^i \in K[Y]. \quad (4.2)$$

define its λ -norm $\|A(Y)\|_\lambda$ to be $\max_i |a_i| \lambda^i$. If λ is understood we shall write simply $\|A(Y)\|$ instead of $\|A(Y)\|_\lambda$.

When $A(Y)$ is the constant polynomial a_0 , that is, when $n = 0$, then $\|A(Y)\|_\lambda = |a_0|$, independent of λ . Suppose $\lambda = |\pi|^{s/r}$, then, $\|aX^r\|_\lambda = |a\pi^s|$. If $A(Y) = \sum_{i=0}^n a_i Y^i$ is pure (see definition 3.3) with slope $-s/r$ then $\|A(Y)\|_\lambda = |a_0|$.

Lemma 4.3. Suppose that

1. $A(Y) = \sum_{i=0}^k a_i Y^i$ is a polynomial in $K[Y]$ of degree k ;
2. $B(Y) = \sum_{i=0}^l b_i Y^i$ is a non-zero polynomial in $K[Y]$ of degree $l \leq k$;
3. $\|B(Y)\| = \|b_l Y^l\|$; equivalently, $|b_l| \lambda^l = \max_i |b_i| \lambda^i$.

Define $C(Y) = A(Y) - Y^{k-l}(a_k/b_l)B(Y)$. In other words, $C(Y)$ is the first remainder and $(a_k/b_l)Y^{k-l}$ is the first quotient obtained when dividing $A(Y)$ by $B(Y)$ using the classical division algorithm. Then

1. $\|C(Y)\| \leq \|A(Y)\|$, and
2. $\|(a_k/b_l)Y^{k-l}\| \leq \|A(Y)\|/\|B(Y)\|$.

Proof. Define $b_i = 0$ when $i < 0$. Then

$$C(Y) = \sum_{i=1}^k \left(a_{k-i} - \frac{a_k b_{l-i}}{b_l} \right) Y^{k-i}. \quad (4.4)$$

Hence,

$$\begin{aligned} \|C(Y)\| &= \max_{1 \leq i \leq k} \lambda^{k-i} \left| a_{k-i} - \frac{a_k b_{l-i}}{b_l} \right| \\ &\leq \max_{1 \leq i \leq k} \max \left(\lambda^{k-i} |a_{k-i}|, \frac{\lambda^k |a_k| \lambda^{l-i} |b_{l-i}|}{\lambda^l |b_l|} \right) \\ &\leq \max_{0 \leq i \leq k} (\lambda^{k-i} |a_{k-i}|, \lambda^k |a_k|) \\ &= \|A(Y)\|. \end{aligned} \quad (4.5)$$

The remainder of the proof is clear. \square

Lemma 4.6. Suppose that $A(Y)$ and $B(Y)$ are polynomials satisfying hypothesis 1, 2, and 3 of Lemma 4.3. Suppose that $Q(Y)$ and $V(Y)$ are the quotient and remainder, respectively, when $A(Y)$ is divided by $B(Y)$; that is,

$$A(Y) = B(Y)Q(Y) + V(Y), \quad (4.7)$$

where $A(Y)$, $B(Y)$, $Q(Y)$, and $V(Y)$ are polynomials in $K[Y]$ such that $\deg V(Y) < \deg B(Y)$. Then

$$\|V(Y)\| \leq \|A(Y)\| \quad \text{and} \quad \|Q(Y)\| \leq \|A(Y)\|/\|B(Y)\|. \quad (4.8)$$

Proof. Apply Lemma 4.3 repeatedly. \square

Lemma 4.9. Suppose that we are given a 7-tuple

$$(k, \mu, B(Y), C(Y), u(Y), v(Y), \epsilon(Y)) \quad (4.10)$$

where k is a positive integer, where μ is real number ≥ 1 , and where the remaining five entries are polynomials in $K[Y]$. Suppose that the following conditions are satisfied:

1. $B(Y) = \sum_{i=0}^l b_i Y^i$ and $C(Y) = \sum_{i=0}^m c_i Y^i$ are non-zero polynomials in $K[Y]$ of degrees, respectively, l and m , such that

$$\|B(Y)\| = \|b_l Y^l\| = \|C(Y)\| = 1; \quad (4.11)$$

2. $\|u(Y)\| \leq \mu$ and $\|v(Y)\| \leq \mu$;
3. $\|u(Y)B(Y) + v(Y)C(Y) - 1\| < 1$;
4. $\deg \epsilon(Y) \leq k$ and $l + m \leq k$.

Then there exist a pair of polynomials $(U(Y), V(Y))$, each in $K[Y]$, such that:

1. $\|U(Y)\| \leq \mu \|\epsilon(Y)\|$ and $\deg U(Y) \leq k - l$;
2. $\|V(Y)\| \leq \mu \|\epsilon(Y)\|$ and $\deg V(Y) \leq l - 1$;
3. $\|U(Y)B(Y) + V(Y)C(Y) - \epsilon(Y)\| < \|\epsilon(Y)\|$.

Proof. From hypothesis 3 we obtain

$$\|\epsilon(Y)u(Y)B(Y) + \epsilon(Y)v(Y)C(Y) - \epsilon(Y)\| < \|\epsilon(Y)\| \quad (4.12)$$

Let $Q(Y)$ be the quotient and $V(Y)$ be the remainder when $\epsilon(Y)v(Y)$ is divided by $B(Y)$; that is, $\epsilon(Y)v(Y) = Q(Y)B(Y) + V(Y)$, where $Q(Y)$ and $V(Y)$ are polynomials in $K[Y]$ with $\deg V(Y) \leq l - 1$. By Lemma 4.6,

$$\|V(Y)\| \leq \|\epsilon(Y)v(Y)\| \leq \mu \|\epsilon(Y)\| \quad (4.13)$$

and

$$\|Q(Y)\| \leq \|\epsilon(Y)v(Y)\|/\|B(Y)\| \leq \mu\|\epsilon(Y)\| \quad (4.14)$$

Next,

$$\begin{aligned} & \epsilon(Y)u(Y)B(Y) + \epsilon(Y)v(Y)C(Y) - \epsilon(Y) \\ &= \epsilon(Y)u(Y)B(Y) + (Q(Y)B(Y) + V(Y))C(Y) - \epsilon(Y) \\ &= (\epsilon(Y)u(Y) + Q(Y)C(Y))B(Y) + V(Y)C(Y) - \epsilon(Y) \\ &= U'(Y)B(Y) + V(Y)C(Y) - \epsilon(Y), \end{aligned} \quad (4.15)$$

where

$$U'(Y) = \epsilon(Y)u(Y) + Q(Y)C(Y). \quad (4.16)$$

Then,

$$\begin{aligned} \|U'(Y)\| &\leq \max(\|\epsilon(Y)u(Y)\|, \|Q(Y)C(Y)\|) \\ &\leq \mu\|\epsilon(Y)\| \end{aligned} \quad (4.17)$$

and

$$\|U'(Y)B(Y) + V(Y)C(Y) - \epsilon(Y)\| < \|\epsilon(Y)\|. \quad (4.18)$$

The polynomial $V(Y)$ already meets the requirements of the Lemma. We show that we can modify $U'(Y)$ to obtain the required polynomial $U(Y)$. Write

$$U'(Y) = \sum_i u_i Y^i. \quad (4.19)$$

If any monomial $u_i Y^i$ satisfies $\|u_i Y^i\| < \|\epsilon(Y)\|$, then we may replace u_i by 0; this will not affect the validity of (4.18). Define $U(Y)$ to be the polynomial obtained from $U'(Y)$ by replacing all such monomials $u_i Y^i$ by 0. Then,

$$\|U(Y)B(Y) + V(Y)C(Y) - \epsilon(Y)\| < \|\epsilon(Y)\|. \quad (4.20)$$

Put $j = \deg U(Y)$. If $j \leq k - l$, we are done. If not, then, the term of highest degree in the product $U(Y)B(Y)$ has degree $j + l > k$. Since $\deg V(Y)C(Y) \leq l - 1 + m < k$ and $\deg \epsilon(Y) \leq k$, the term of highest degree in the product $U(Y)B(Y)$ must also be the term of highest degree in the left-hand side of $U(Y)B(Y) + V(Y)C(Y) - \epsilon(Y)$. The norm of this term is $\|u_j Y^j\| \|b_l Y^l\| \geq \|\epsilon(Y)\|$. This contradicts (4.20) and shows that $j + l \leq k$, equivalently $\deg U(Y) \leq k - l$. \square

For the remainder of this section we assume that λ is a rational power of $|\pi|$. Specifically, $\lambda = |\pi|^{s/r}$, where r and s are relatively prime integers with $r \geq 1$. In particular, we require that if $s = 0$, then $r = 1$. Under this assumption, the norm $\|A(Y)\|$ of any non-zero polynomial $A(Y) \in K[Y]$ will be an integral power of $|\pi|^{1/r}$.

We can now state the form of Hensel's Lemma that we use.

Theorem 4.21. (Hensel's Lemma) Suppose that h is a non-negative integer and that we are given a 5-tuple of polynomials

$$(R(Y), B_0(Y), C_0(Y), u(Y), v(Y)) \quad (4.22)$$

each with coefficients in K such that

1. $R(Y)$ has degree k and satisfies $\|R(Y)\| = 1$;
2. $B_0(Y) = \sum_{i=0}^l b_i Y^i$ has degree l and satisfies $\|B_0(Y)\| = \|b_l Y^l\| = 1$;
3. $C_0(Y) = \sum_{i=0}^m c_i Y^i$ has degree m and satisfies $\|C_0(Y)\| = 1$;
4. $\|R(Y) - B_0(Y)C_0(Y)\| \leq |\pi|^{(2h+1)/r}$;
5. $\|u(Y)\| \leq |\pi|^{-h/r}$, $\|v(Y)\| \leq |\pi|^{-h/r}$;
6. $\|u(Y)B_0(Y) + v(Y)C_0(Y) - 1\| < 1$.

Then there exist polynomials $B(Y)$ and $C(Y)$ in $K[Y]$ such that

1. $R(Y) = B(Y)C(Y)$;
2. $\|B(Y) - B_0(Y)\| < |\pi|^{h/r}$;
3. $\|C(Y) - C_0(Y)\| < |\pi|^{h/r}$;
4. $\deg B(Y) = \deg B_0(Y)$.

Proof. We first show that we may assume that $k \geq m + l$. If $k < l + m$, then the term of highest degree of $R(Y) - B_0(Y)C_0(Y)$ is $-b_l c_m Y^{m+1}$ whose norm, by hypotheses (2) and (4), satisfies

$$\|b_l Y^l\| \|c_m Y^m\| = \|b_l c_m Y^{l+m}\| \leq |\pi|^{(2h+1)/r}, \quad (4.23)$$

so that $\|c_m Y^m\| \leq |\pi|^{(2h+1)/r}$. It follows that if we replace $C_0(Y)$ by the lower degree polynomial $C_0(Y) - c_m Y^m$ and replace m by the degree of this new $C_0(Y)$, then the hypotheses remain satisfied. For the remainder of this proof we assume that $k \geq l + m$.

We shall construct sequences of polynomials $\{B_i(Y)\}$ and $\{C_i(Y)\}$ for $i = 1, 2, \dots$ such that

1. $\|B_i(Y) - B_{i-1}(Y)\| \leq |\pi|^{(h+i)/r}$ and $\deg B_i(Y) = l$;
2. $\|C_i(Y) - C_{i-1}(Y)\| \leq |\pi|^{(h+i)/r}$ and $\deg C_i(Y) \leq m - l$;
3. $\|R(Y) - B_i(Y)C_i(Y)\| \leq |\pi|^{(2h+i+1)/r}$.

Putting $B(Y) = \lim_{i \rightarrow \infty} B_i(Y)$ and $C(Y) = \lim_{i \rightarrow \infty} C_i(Y)$ will complete the proof.

We proceed by induction on the variable i , starting with $i = 1$. Put $\epsilon_i(Y) = R(Y) - B_{i-1}(Y)C_{i-1}(Y)$ so that, by hypothesis (when $i = 1$) or induction (when $i > 1$), $\|\epsilon_i(Y)\| \leq |\pi|^{(2h+i)/r}$. Apply Lemma 4.9 to the 7-tuple

$$(k, |\pi|^{-h}, B_i(Y), C_i(Y), u(Y), v(Y), \epsilon_i(Y)). \quad (4.24)$$

Lemma 4.9 returns a pair of polynomials which we denote $(U_i(Y), V_i(U))$. These polynomials satisfy

1. $\|U_i(Y)\| \leq |\pi|^{(h+i)/r}$ and $\deg U_i(Y) \leq m - 1$;
2. $\|V_i(Y)\| \leq |\pi|^{(h+i)/r}$ and $\deg V_i(Y) \leq l - 1$;
3. $\|U_i(Y)B_0(Y) + V_i(Y)C_0(Y) - \epsilon_i(Y)\| \leq |\pi|^{(2h+i+1)/r}$.

Define

$$B_i(Y) = B_{i-1}(Y) + V_i(Y), \quad C_i(Y) = C_{i-1}(Y) + U_i(Y) \quad (4.25)$$

Then

$$\begin{aligned} \|R(Y) - B_i(Y)C_i(Y)\| &= \|R(Y) - (B_{i-1}(Y) + V_i(Y))(C_{i-1}(Y) + U_i(Y))\| \\ &= \|(R(Y) - B_{i-1}(Y)C_{i-1}(Y)) \\ &\quad - (U_i(Y)B_{i-1}(Y) + V_i(Y)C_{i-1}(Y)) - U_i(Y)V_i(Y)\| \\ &= \|(\epsilon_i(Y) - (U_i(Y)B_{i-1}(Y) + V_i(Y)C_{i-1}(Y))) \\ &\quad - U_i(Y)V_i(Y)\| \\ &\leq \max(|\pi|^{2h+1}, |\pi|^{2h+2i}) \\ &= |\pi|^{(2h+i+1)/r} \end{aligned} \quad (4.26)$$

□

The proof of Hensel's Lemma consists of an algorithm. If only approximations to the factors $R(Y)$ and $B(Y)$ are needed, then the algorithm is finite. We shall call the algorithm *Hensel's Lemma*, also.

Now suppose that we are given a polynomial $R(Y)$ which is pure and whose Newton diagram has slope $-s/r$, where r and s are relatively prime integers with $r > 0$. The degree of $R(Y)$ must be a multiple of r , say kr . Both of the points $(0, \text{ord } a_0)$ and $(kr, \text{ord } a_{kr})$ must lie on this segment. We can write

$$R(Y) = \sum_{i=0}^k a_i \pi^{-is} Y^{ir} + [\text{terms above the Newton diagram}] \quad (4.27)$$

where $|a_i| \leq 1$ for $0 \leq i \leq k$, and where, In the $\lambda = |\pi|^{s/r}$ norm,

$$\|R(Y)\| = |a_0| = \|a_{kr} Y^{kr}\| = |a_k|. \quad (4.28)$$

Equation (4.27) can be restated as

$$\|R(Y) - \sum_{i=0}^k a_i \pi^{-is} Y^{ir}\| < \|R(Y)\|. \quad (4.29)$$

When this is the case we have

Corollary 4.30. Suppose that $R(Y)$ is a pure polynomial of degree kr , of the form (4.27) which satisfies (4.28) and suppose further that the polynomial

$R^*(Y) = \sum_{i=0}^k \bar{a}_i Y^i$ satisfies $R^*(Y) = \beta(Y)\gamma(Y)$ where $\beta(Y)$ and $\gamma(Y)$ are monic, relatively prime polynomials in $\overline{K}[Y]$. Then $R(Y) = B(Y)C(Y)$ where $B(Y)$ and $C(Y)$ are relatively prime polynomials in $K[Y]$ satisfying $B^*(Y) = \beta(Y)$ and $C^*(Y) = \gamma(Y)$.

Proof. By multiplying $R(Y)$ by an appropriate power of π , we may assume that $\|R(Y)\| = 1$. Suppose that $\deg \beta(Y) = l$ and $\deg \gamma(Y) = m$. There exist polynomials $\mu(Y)$ and $\nu(Y)$ in $\overline{K}[Y]$ such that $\mu(Y)\beta(Y) + \nu(Y)\gamma(Y) = 1$ and such that $\deg \mu(Y) < m$ and $\deg \nu(Y) < l$. Choose elements b_i, c_i, u_i , and v_i in K such that

$$\begin{aligned}\beta(Y) &= \sum_{i=0}^l \bar{b}_i Y^i, & \gamma(Y) &= \sum_{i=0}^m \bar{c}_i Y^i, \\ \mu(Y) &= \sum_{i=0}^{m-1} \bar{u}_i Y^i, & \nu(Y) &= \sum_{i=0}^{l-1} \bar{v}_i Y^i.\end{aligned}\tag{4.31}$$

Define

$$\begin{aligned}B_0(Y) &= \sum_{i=0}^l b_i \pi^{-is} Y^{ir}, & C_0(Y) &= \sum_{i=0}^m c_i \pi^{-is} Y^{ir}, \\ u(Y) &= \sum_{i=0}^{m-1} u_i \pi^{-is} Y^{ir}, & v(Y) &= \sum_{i=0}^{l-1} v_i \pi^{-is} Y^{ir}.\end{aligned}\tag{4.32}$$

Then $B_0(Y)^* = \beta(Y)$, $C_0(Y)^* = \gamma(Y)$, $u(Y)^* = \mu(Y)$ and $v(Y)^* = \nu(Y)$. Apply Theorem 4.21 with $h = 0$ to the 5-tuple

$$(R(Y), B_0(Y), C_0(Y), u(Y), v(Y)).\tag{4.33}$$

The result will be two polynomials $B(Y)$ and $C(Y)$ which meet the requirements of this corollary. \square

The special case of this Corollary when $C(Y)$ is pure with horizontal Newton diagram appears as Lemma 4.1 in [6].

4.2 Hensel Factor

We may now define **Hensel Factor**, an important subroutine of our algorithm. It takes as input a triple $(K, F(X), A(X))$, where K is a field, $F(X)$ is a polynomial of degree ≥ 2 to be factored, and $A(X)$ is a non-zero polynomial of degree $< \deg F(X)$. We will say the algorithm *succeeds* if one of Lemmas 3.4, 3.5 or Corollary 4.30 apply. If Lemma 3.5 holds, then $(K, F(X), A(X))$ forms a certificate for the irreducibility of $F(X)$, and we are done. If Lemma 3.4 or Corollary 4.30 hold, then we have found a factor $G(X)$ of $F(X)$ over a field L , and we recursively call **p-adic Factor** with input $(L, G(X))$. If none of the lemmas apply, we say it *fails*.

Hensel Factor. Input $(K, F(X), A(X))$.

1. Compute $R(Y) = \text{Res}_X(F(X), Y - A(X))$.

Comment. Each of the elements $A(x)$, where x is a root of $F(X)$, is a root of $R(Y)$. If the resultant $R(Y)$ were a monomial, then the n distinct roots x of $F(X)$ would satisfy the polynomial $A(X)$, of degree $< n$. Thus $R(Y)$ is not a monomial.

2. There are now four sub-cases, at most one of which can hold:

- (a) The polynomial $R(Y)$ is not pure.

Factor $R(Y)$ using Lemma 3.4. Then factor $F(X)$ using Lemma 2.3. Let $G(X)$ be a factor of least degree. Restart **p-adic Factor** with the pair $(K, G(X))$.

- (b) The polynomial $R(Y)$ is pure and $R^*(Y)$ can be written as a product of two relatively prime factors, each of degree ≥ 1 in $\bar{K}[X]$.

Factor $R(Y)$ using Corollary 4.30 of Hensel's Lemma. Then factor $F(X)$ using Lemma 2.3. Let $G(X)$ be a factor of least degree. Restart **p-adic Factor** with the pair $(K, G(X))$.

- (c) The polynomial $R(Y)$ is pure and $R^*(Y)$ is the e^{th} power of an irreducible monic polynomial $\alpha(Y)$ of degree ≥ 2 in $\bar{K}[Y]$.

Choose a polynomial $u(Y) \in K[Y]$ such that $\bar{u}(Y) = \alpha(Y)$. Denote by L the unramified extension field of K obtained by adjoining a root y of $u(Y)$ to K . Put $\beta(Y) = (Y - \bar{y})^e$ and put $\gamma(Y) = R^*(Y)/\beta(Y)$. Then $R^*(Y) = \beta(Y)\gamma(Y)$ where $(\beta(Y), \gamma(Y)) = 1$. By Corollary 4.30 we can factor $R(Y)$ as $R(Y) = B(Y)C(Y)$ where $B^*(Y) = \beta(Y)$. Factor $F(X)$ over L using Lemma 2.3 with $R_1(Y) = B(Y)$ and $R_2(Y) = C(Y)$. Let $F_1(X)$ be the factor of $F(X)$ corresponding to $R_1(Y)$. Restart **p-adic Factor** with the pair $(L, F_1(X))$.

Comment. Note that the field L is determined uniquely by K and $\alpha(Y)$; it is independent of the specific choice of $u(Y)$ (see Artin [1, page 69, Theorem 2A]). Moreover, if x is a root of $F_1(X)$ in \tilde{K} , then $\bar{y} = \overline{F_1(x)}$. Hence the field L is contained in the field $K(x)$.

- (d) The polynomial $R(Y)$ is pure and the slope of its Newton diagram is k/n where $(k, n) = 1$.

By Lemma 3.5, $F(X)$ is irreducible and the algorithm terminates with the triple $(K, F(X), A(X))$.

3. None of the four cases (2a), (2b), (2c), or (2d) applies, so that $R^*(Y)$ is a power of a linear factor in $\bar{K}[Y]$.

Return **failure**

5 Some Technical Lemmas

We state here some simple results which will be used in the next section. We first have a lemma from elementary number theory. Its proof is constructive.

Lemma 5.1. Suppose that h is a positive integer and that for $1 \leq j \leq h$ we are given fractions s_j/r_j where r_j and s_j are relatively prime positive integers. Define $t_0 = 1$ and for $1 \leq j \leq h$, define $t_j = \text{lcm}(r_1, r_2, \dots, r_j)$. Then, for any integer u , there exist integers e_j , for $1 \leq j \leq h$, satisfying $0 \leq e_j < t_j/t_{j-1}$ and such that

$$\sum_{j=1}^h e_j s_j / r_j - u/t_h \quad (5.2)$$

is an integer.

Proof. The proof proceeds by induction on h . When $h = 1$, then $t_1 = r_1$, and the unique choice for e_1 is the least non-negative, integral solution to $e_1 s_1 \equiv u \pmod{r_1}$.

Suppose that $h > 1$. We will show that there exist integers v and e_h such that $0 \leq e_h < t_h/t_{h-1}$ and such that

$$e_h s_h / r_h + v/t_{h-1} - u/t_h \quad (5.3)$$

is an integer. This will reduce the problem to the $h - 1$ case with u replaced by v . Multiplying (5.3) by t_h shows that we must choose e_h and v to satisfy

$$e_h s_h t_h / r_h + v t_h / t_{h-1} \equiv u \pmod{t_h} \quad (5.4)$$

Now suppose that p is a prime dividing t_h , that $p^\alpha \parallel r_h$ (this means that p^α is the exact power of p dividing r_h), and that $p^\beta \parallel t_{h-1}$. Put $\gamma = \max(\alpha, \beta)$. Since $t_h = \text{lcm}(t_{h-1}, r_h)$, we see that $p^\gamma \parallel t_h$. Then $p^{\gamma-\alpha} \parallel (t_h/r_h)$ and $p^{\gamma-\beta} \parallel (t_h/t_{h-1})$. If $\alpha = \gamma$, then p divides r_h , hence does not divide s_h , so that p does not divide $s_h t_h / r_h$. If $\beta = \gamma$, then p does not divide t_h/t_{h-1} . Thus p divides at most one of $s_h t_h / r_h$ and t_h/t_{h-1} . It follows that $s_h t_h / r_h$ and t_h/t_{h-1} are relatively prime. Hence there exists a solution e_h and v to (5.4) (even with equality replacing congruence). For any integer k the pair $(e_h + kt_h/t_{h-1}, v - ks_h t_h / r_h)$ is also a solution of (5.4). Replacing e_h by $e_h + kt_h/t_{h-1}$ for an appropriate integer k allows us to choose e_h to satisfy $0 \leq e_h < t_h/t_{h-1}$. \square

This immediately gives the following corollary, which will be used in the algorithm to construct a polynomial $E(X)$ with specified values of $E(x)$ for the roots x of $F(X)$.

Corollary 5.5. Suppose that h , the fractions s_j/r_j and the integers t_j satisfy the hypotheses of Lemma 5.1. Suppose that A_1, A_2, \dots, A_h are elements of $\tilde{\mathbb{Q}}_p$ such that $\text{ord } A_j = s_j/r_j$. Then for any integer u there exist integers e_1, e_2, \dots, e_h satisfying $0 \leq e_j < t_j/t_{j-1}$ and an integer e_0 such that $\text{ord } \pi^{e_0} \prod_{j=1}^h A_j^{e_j} = u/t$.

The next lemma shows that if a monic polynomial of degree m is “small” at $n > m$ distinct points, then at least two of these points must be “close” to each other. If the points are given in advance, then there is a limit to how “small” the polynomial can be at all n points.

Lemma 5.6. Suppose that x_1, x_2, \dots, x_n are elements of \tilde{K} and that $A(X)$ is a monic polynomial in $\tilde{K}[X]$ of degree $m < n$. Then $\min_{j \neq j'} |x_j - x_{j'}|^m \leq \max_i |A(x_i)|$.

Proof. Put $\epsilon = \max_j |A(x_j)|$. We can write $A(X) = \prod_{i=1}^m (X - \theta_i)$ where the $\theta_i \in \tilde{\mathbb{Q}}_p$ are the roots of $A(X)$. Then for each j ,

$$\epsilon \geq |A(x_j)| = \prod_{i=1}^m |x_j - \theta_i|. \quad (5.7)$$

Not all of the factors $|x_j - \theta_i|$ on the right-hand side of (5.7) can be $> \epsilon^{1/m}$. Hence there must exist a value of i , call it $\sigma(j)$, such that $|x_j - \theta_{\sigma(j)}| \leq \epsilon^{1/m}$. By doing this for all j , we obtain a map σ from the set $\{1, 2, \dots, n\}$ to the set $\{1, 2, \dots, m\}$. Since $n > m$, there must be two values, $j \neq j'$ such that $\sigma(j) = \sigma(j')$. Call this common value k . Then both $|x_j - \sigma_k| \leq \epsilon^{1/m}$ and $|x_{j'} - \sigma_k| \leq \epsilon^{1/m}$. Hence $|x_j - x_{j'}| \leq \epsilon^{1/m}$ \square

Corollary 5.8. Suppose that $F(X)$ is a monic polynomial in $O_K[X]$ of degree n with distinct roots x_1, x_2, \dots, x_n . If $A(X)$ is a monic polynomial in $K[X]$ of degree $m < n$, then, for at least one i , we have $|A(x_i)| \geq |\Delta_F|^m$.

Proof. Because all $|x_i| \leq 1$, we have

$$\Delta_F = \prod_{i \neq j} |x_i - x_j| \leq \min_{i \neq j} |x_i - x_j| \quad (5.9)$$

Now apply Lemma 5.6. \square

6 The p -Adic Factor Algorithm

In this section, we describe the main algorithm. It will find an irreducible factor $H(X)$ of $F(X)$ along with a certificate that $H(X)$ is irreducible. To completely factor $F(X)$, the algorithm may have to be repeated, perhaps several times, with $F(X)/H(X)$ replacing $F(X)$ until this quotient is 1.

The algorithm will attempt to factor $F(X)$ using **Hensel Factor** with $A(X) = X$. This will fail only when $F^*(X)$ has the form $(X - \alpha)^m$. When this occurs, the algorithm will systematically look for a polynomial $A(X) \in K[X]$ for which **Hensel Factor** succeeds.

Because the algorithm is recursive and both the polynomial to be factored and the local field may change during the course of the algorithm we will, for the

remainder of this paper, denote by $F_0(X)$ the original polynomial to be factored over the original field K_0 .

The input to the algorithm is a pair $(K, F(X))$, where K is either K_0 or a finite, unramified extension of K_0 , and $F(X)$ is a monic polynomial of degree $n \geq 2$ with coefficients in O_K dividing $F_0(X)$. We assume $F(X)$ has no multiple factors and $F(0) \neq 0$. Since we compute approximations to the factors, $F(X)$ will not in general be known exactly. In Section 8 we determine how much precision is needed to avoid errors in the factorization.

The **p -adic Factor** algorithm will return a field L which is an unramified extension of K of degree $\leq n$, a polynomial $G(X)$ in $L[X]$ dividing $F(X)$, and a polynomial $B(X) \in L[X]$ of degree $< \deg G(X)$. By Lemma 3.5, the triple $(L, G(X), B(X))$ provides the proof that $G(X)$ is irreducible.

By Lemma 2.9,

$$H(X) = \text{Norm}_{L/K} G(X). \quad (6.1)$$

is an irreducible factor of $F(X)$. As noted above, the algorithm may then be called recursively on the pair $(K, F(X)/H(X))$ to complete the factorization of $F(X)$.

Section 6.1 presents the algorithm, after which Section 6.2 describes in more detail what certain steps are doing, and why they work.

6.1 The Algorithm

p -adic Factor. Input: $(K, F(X))$.

Step 1. Apply **Hensel Factor** to $(K, F(X), X)$ (in this case $\text{Res}_X(F(X), Y - X) = F(Y)$).

Step 2. We reach this step only if **Hensel Factor** did not succeed in Step 1, so $F^*(X)$ is a power of a linear polynomial. Choose $\alpha \in \mathcal{A}$ such that

$$F(X) = (X^r - \alpha\pi^s)^m + [\text{terms above the Newton diagram}] \quad (6.2)$$

where

- (a) $\bar{\alpha}$ is the unique root of $F^*(X)$ in \overline{K} and $\text{ord } \alpha = 0$;
- (b) $r < n$ and $m > 1$;
- (c) $mr = n$; $\gcd(r, s) = 1$;
- (d) the Newton diagram of $F(X)$ has slope $-s/r$.

Step 3. We initiate the outer loop by putting $A_1(X) = X$, $R_1(Y) = F(Y)$, $r_1 = r$, $s_1 = s$, $t_0 = 1$, and $t_1 = r_1$.

Step 4. (Outer loop) For $h = 1, 2, \dots$, perform Steps 5 through 11.

Step 5. To begin the inner loop, put

$$\begin{aligned} B_0(X) &= A_h(X)^{t_h/t_{h-1}}, \\ S_0(Y) &= \text{Res}_X(F(X), Y - B_0(X)), \\ u_0 &= s_h t_h^2 / (r_h t_{h-1}). \end{aligned}$$

Step 6. (Inner Loop) For $i = 0, 1, \dots$, perform Steps 7 through 10.

Step 7. Use Corollary 5.5 to choose integers e_j , for $0 \leq j \leq h$, such that

- (a) $0 \leq e_j \leq t_j/t_{j-1} - 1$ when $1 \leq j \leq h$,
- (b) $e_0 + \sum_{j=1}^h e_j s_j/r_j = u_i/t_h$ (in the notation of Corollary 5.5, $e_0 = u/t_h - \sum_{j=1}^h e_j s_j/r_j$).

Define a polynomial $E(X)$ by

$$E(X) = \pi^{e_0} A_1(X)^{e_1} A_2(x)^{e_2} \cdots A_h(X)^{e_h}. \quad (6.3)$$

Step 8. Define

$$C(X) = B_i(X) E(X)^{-1} \pmod{F(X)} \quad (6.4)$$

and

$$T(Y) = \text{Res}_X(F(X), Y - C(X)). \quad (6.5)$$

Apply **Hensel Factor** to the triple $(K, F(X), C(X))$.

Step 9. Put $B(X) = B_i(X) - \alpha E(X)$ and $S(Y) = \text{Res}_X(F(X), Y - B(X))$. Apply **Hensel Factor** to the triple $(K, F(X), B(X))$.

Step 10. If the common value $\text{ord } B(x)$ can be written in the form u/t_h , where u is an integer, then put $B_{i+1}(X) = B(X)$, $S_{i+1}(Y) = S(Y)$, $u_{i+1} = u$. and continue the “inner loop” by returning to Step 6.

Step 11. Denote the common value of $\text{ord } B(x)$ by s_{h+1}/r_{h+1} , where r_{h+1} and s_{h+1} are relatively prime, non-negative integers as before. Put $A_{h+1}(X) = B(X)$, $R_{h+1}(Y) = S(Y)$, and $t_{h+1} = \text{lcm}(t_h, r_{h+1})$.

- (a) If $t_{h+1} < n$ continue the “outer loop” by returning to Step 4, with h increased by 1.
- (b) Otherwise use Corollary 5.5 to choose integers e_j for $0 \leq j \leq h+1$ such that

- i. $0 \leq e_j \leq t_j/t_{j-1} - 1$ when $1 \leq j \leq h$ and
- ii. $\sum_{j=1}^{h+1} e_j s_j/r_j - 1/t_{h+1} = e_0$;

Define $E(X)$ by

$$E(X) = \pi^{e_0} A_1(X)^{e_1} A_2(x)^{e_2} \cdots A_h(X)^{e_h} \quad (6.6)$$

and apply **Hensel Factor** to the triple $(K, F(X), E(X))$.

6.2 Discussion of the Algorithm

In Step 2, each (unknown) root x of $F(X)$ has $\text{ord } x = s/r$ by (6.2). This shows that the ramification index of each of the n field extensions of the form $K(x)/K$ is divisible by r .

Starting with $A_1(X) = X$ at Step 3, the outer loop defines a finite sequence of polynomials $A_1(X), A_2(X), \dots$ and a corresponding sequence of pairs of non-negative integers, $(r_1, s_1), (r_2, s_2), \dots$, where each of the pairs (r_i, s_i) are relatively prime. We have $R_h(Y) = \text{Res}_X(F(X), Y - A_h(X))$, $t_0 = 1$, and for $h \geq 0$, define $t_h = \text{lcm}(r_1, r_2, \dots, r_h)$. The the following properties are easily checked:

1. Each of the r_h and each of the t_h divides n .
2. The polynomial $A_h(X)$ is monic of degree t_{h-1} .
3. There exists an element $\alpha \in \mathcal{A}$ such that $\text{ord } \alpha = 0$ and

$$R_h(Y) = (Y^{r_h} - \alpha\pi^{s_h})^{n/r_h} + [\text{terms above the Newton diagram}].$$

It follows that for each root x of $F(X)$, we have

$$\text{ord } A_h(x) = s_h/r_h. \quad (6.7)$$

Thus the multiplicative group generated by $|\pi|, |A_1(x)|, |A_2(x)|, \dots, |A_h(x)|$ is independent of the choice of x and contains the value group of K . Hence, for each root x of $F(x)$, the ramification index of the field extension $K(x)/K$ is divisible by r_h .

4. The integer r_h does not divide t_{h-1} and for each root x of $F(X)$, the ramification index of the field extension $K(x)/K$ is divisible by t_h . It follows that $t_1 < t_2 < \dots < t_h \leq n$.

Since t_i is a proper divisor of t_{i+1} , we must have $h \leq \log_2 n$. This limits the number of steps of the outer loop.

To determine $A_{h+1}(X)$, we attempt in the inner loop to find a monic polynomial $B(X)$ of degree t_h satisfied by all roots x of $F(X)$. Since $F(X)$ has $n > t_h$ distinct roots, this attempt must fail. Its failure either leads to a situation where we can factor $F(X)$ using Hensel's lemma or leads to the determination of $A_{h+1}(X)$. The inner loop finds $A_{h+1}(X)$ by defining a sequence of polynomials

$$B_0(X), B_1(X), B_2(X), \dots \quad (6.8)$$

and a corresponding, strictly increasing, finite sequence of non-negative integers $u_0 < u_1 < u_2, \dots$

Each polynomial $B_i(X)$ is monic of degree t_h . Each root x of $F(X)$ will satisfy $\text{ord } B_i(x) = u_i/t_h$. Corollary 5.8 provides an upper bound for u_i and hence the sequence $B_0(X), B_1(X), \dots$ will be finite.

In Step 7, we have constructed $E(X)$ so that $\text{ord } E(x) = u_i/t_h$ for every root x of $F(X)$. Since $\deg A_j(X) \leq t_{j-1}$, we obtain, from Step 7a, have

$$\begin{aligned} \deg E(X) &\leq \sum_{j=1}^h (t_j/t_{j-1} - 1)t_{j-1} \\ &= \sum_{j=1}^h (t_j - t_{j-1}) \\ &= t_h - 1. \end{aligned} \quad (6.9)$$

In Step 8, (6.4) is valid because $E(X)$ and $F(X)$ have no common zeros. The polynomial $T(Y)$ is monic of degree n and, for each root x of $F(X)$, we have $|C(x)| = |B_i(x)/E(x)| = 1$. Consequently, the Newton diagram of $T(Y)$ is the horizontal line-segment connecting the points $(0, 0)$ and $(n, 0)$. It follows that

the polynomial $T^*(Y)$ is monic of degree n and its constant term is not zero. If **Hensel Factor** fails, then we can write

$$T(Y) = (Y - \alpha)^n + [\text{terms above the Newton diagram}] \quad (6.10)$$

where $\alpha \in \mathcal{A}$ and $\text{ord } \alpha = 0$.

After Step 10, since $B_i(X)$ is monic of degree t_h and $\deg E(X) < t_h$, $B(X)$ is monic of degree t_h . By the definition of α , we have $\text{ord } B_i(x) - \alpha E(x) > 0$ for each root x of $F(X)$. It follows that $\text{ord } B(x) > \text{ord } B_i(X)$ for each such x . If **Hensel Factor** fails, then $\text{ord } B(x)$ is the same for all roots x of $F(X)$ and is $> u_i/t_h$.

Put $\delta = \text{ord } |\Delta_F|$. Step 6 will increase i by 1. Since t_h divides n and the u_i are non-negative integers and strictly increasing we have $u_i/t_h \geq i/n$. By Corollary 5.8, we see that $u_i/t_h \leq \delta n$. Thus $i \leq \delta n^2$. This means that for each value of h , the inner-loop is performed at most δn^2 times.

In Step 11a, r_{h+1} does not divide t_h , so that $t_{h+1} > t_h$. In Step 11b, we have $\text{ord } E(x) = 1/n$ for every root x of $F(X)$, so case 2d of **Hensel Factor** will succeed, and this will lead to finding an irreducible factor of $F_0(X)$.

7 Two Examples

We decided to implement the algorithm, both to verify its correctness and practicality, and to allow experimentation. The first decision was to choose a mathematical package in which to implement it. MAGMA [4] was the original choice, but a package to perform local field operations was delayed several times, so the implementation was done in GP instead. GP is a part of the PARI system developed by Henri Cohen [2]. It does support p -adic fields, and is flexible enough to support unramified extension fields of the p -adics relatively easily. A new version of MAGMA with local fields has recently appeared, so a port of the algorithm to MAGMA is planned.

The resulting code is available at the second author's web site [13]. Because of the overhead of GP, it is slower than the PARI routine `factorpadic` for most polynomials. An implementation in C using the PARI library would run in about the same time as `factorpadic` for most polynomials.

For an example of how the algorithm functions, we will factor the polynomial

$$F(X) = (X - 4)^2(X^2 - 2) + 2^{100} \quad (7.1)$$

over \mathbb{Q}_2 .

If we apply **p-adic Factor** to this polynomial, it starts by attempting to apply **Hensel Factor**. The Newton diagram of $R(Y) = F(Y)$ is not pure, so using Hensel's Lemma we find factors

$$G_1(X) = (X^2 - 2) + (2^{101} + 2^{105} + \dots)X + (2^{99} + 2^{102} + \dots) \quad (7.2)$$

and

$$G_2(X) = (X - 4)^2 + (2^{101} + 2^{102} + \dots)X + (2^{99} + 2^{100} + \dots) \quad (7.3)$$

Attempting to factor $G_1(X)$, we call **Hensel Factor** again. This time, the Newton diagram is pure, and we are in subcase (2d). Thus $G_1(X)$ is irreducible.

$G_2(X)$ is also pure, but its slope and degree are both even, so **Hensel Factor** does not apply. We have $G_2^*(X) = (X - 1)^2$.

In Step 2 of **p-adic Factor**, we have $\alpha = 1$, $r = 1$, $s = 2$, and $n = m = 2$. We arrive in Step 7 with $E(X) = 4$, $C(X) = X/4$, and

$$T(Y) = Y^2 - 2Y + (1 + 2^{95} + \dots). \quad (7.4)$$

The Newton diagram of $T(Y)$ is now horizontal, but $T^*(Y) = (Y - 1)^2$ is still a power of a linear polynomial, so the call to **Hensel Factor** in Step 8 fails.

In Step 9, we have $\alpha = 1$ and $B(X) = X - 4$. This gives

$$S(Y) = Y^2 + (2^{101} + \dots)Y + (2^{99} + \dots). \quad (7.5)$$

The call to **Hensel Factor** in Step 9 now goes to subcase (2d), and we have proved that $G_2(X)$ is irreducible, completing the factorization of $F(X)$.

Very few polynomials make it all the way through the inner loop more than once. One that does is

$$F(X) = (X^2 - 2 - 2^{20})(X^2 - 2 + 2^{20}) \quad (7.6)$$

over \mathbb{Q}_2 .

We have $F^*(X) = (X - 1)^2$, so **Hensel Factor** fails. In Step 2 we choose $\alpha = 1$, $r = 2$, $s = 1$, $m = 2$, and $n = 1$. Entering the inner loop, we find $E(X) = 2$, $C(X) = X^2/2$, and

$$T(Y) = Y^4 - 4Y^3 + (6 - 2^{39})Y^2 + (-4 + 2^{40})Y + (1 - 2^{39} + 2^{76}). \quad (7.7)$$

Again, **Hensel Factor** fails. In Step 9 we set $B(X) = X^2 - 2$, and have

$$S(Y) = Y^4 - 2^{41}Y^2 + 2^{80}. \quad (7.8)$$

Hensel Factor fails on $S(Y)$, and $\text{ord } B(x) = 20$ for each root x of $F(X)$, so we continue the inner loop. Returning to Step 7, we have $E(X) = 2^{20}$, $C(X) = 2^{-20}X^2 - 2^{-19}$, and $T(Y) = Y^4 - 2Y^2 + 1$. Once again, **Hensel Factor** fails.

Finally, we succeed in Step 9. This time we have $B(X) = X^2 - 2 - 2^{20}$, and $S(Y) = Y^4 + 2^{22}Y^3 + 2^{42}Y^2$. The factor of Y^2 in $S(Y)$ yields the factor

$$G_1(X) = X^2 - 2 - 2^{20}. \quad (7.9)$$

Both this factor and the other one immediately are shown to be irreducible by subcase (2d) of **Hensel Factor**.

8 Bounds on Required Precision and Complexity

From the discussion in Section 6.2, it is clear that the loops of **p-adic Factor** will be executed a polynomial number of times in n and $\log |\Delta_F|$. Therefore, to

show that **p-adic Factor** is a random polynomial-time algorithm, we only need to bound the precision needed in the computations.

In general, we can only approximately compute the factors of the p -adic polynomial $F(X)$. This causes two problems. First, in the gcd computation in Lemma 2.3:

$$F_i(X) = \gcd(F(X), R_i(A(X))), \quad (8.1)$$

we do not know the R_i exactly, and so terms in the computation that appear to be zero may not be. In this situation it is difficult to give a reasonable a priori estimate of the accuracy of $R_i(Y)$ that is needed to compute the gcd to the desired accuracy.

To circumvent this difficulty, we give an alternative method of computing $F_i(X)$, which involves solving a system of linear equations.

Lemma 8.2. Suppose that $F(X) \in K[X]$ is a monic polynomial of degree n with distinct roots x_1, x_2, \dots, x_n in the algebraic closure \overline{K} of K . Suppose that $A(X) \in K[X]$. Put $y_i = A(x_i)$, and suppose that the y_i are distinct. Put $R(Y) = \text{Res}_X(F(X), Y - A(X))$. Then there exists a polynomial $B(X) \in K[Y]$ of degree $\leq n - 1$ such that $B(A(X)) \equiv X \pmod{F(X)}$. Furthermore, if $R(Y) = R_1(Y)R_2(Y)$ is a nontrivial factorization of $R(Y)$, then $F(X) = F_1(X)F_2(X)$ where $F_i(X) = \text{Res}_Y(R_i(Y), X - B(Y))$. Finally, $\deg F_i(X) = \deg R_i(Y)$.

Proof. We first show that the n polynomials $A(X)^k \pmod{F}(X)$ for $0 \leq k \leq n - 1$ are linearly independent over K . Suppose that we have a relation

$$\sum_{i=0}^{n-1} b_i A(X)^i \equiv 0 \pmod{F(X)}. \quad (8.3)$$

Substituting the values $x = x_k$ into (8.3) yields the system of linear equations

$$\sum_{i=0}^{n-1} b_i y_k^i = 0 \quad \text{for } 1 \leq i \leq n. \quad (8.4)$$

The matrix of the equations (8.4) is a Vandermonde. Since the y_k are distinct it is nonsingular. This shows that all of the b_i are zero. It follows that the equation

$$\sum_{i=0}^{n-1} b_i A(X)^i \equiv X \pmod{F(X)} \quad (8.5)$$

has a unique solution b_0, b_1, \dots, b_{n-1} . Put $B(Y) = \sum_{i=0}^{n-1} b_i Y^i$. Then

$$B(A(X)) = \sum_{i=0}^{n-1} b_i A(X)^i \equiv X \pmod{F(X)}. \quad (8.6)$$

Suppose that $\deg R_1(Y) = r$. By renumbering we may suppose that the roots of $R_1(Y)$ are y_1, y_2, \dots, y_r where $r < n$. The roots x of $F_1(X)$ are those x for which there exists y such that $R(y) = 0$ and $x - B(y) = 0$. Thus the roots of $F_1(X)$ are x_1, x_2, \dots, x_r where $x_i = B(y_i)$. This shows that $F_1(X)$ is a factor of $F(X)$ of degree r . Similarly, $F_2(X)$ is a factor of $F(X)$ of degree $n - r$. It is immediate from the definition of resultant that $\deg F_i(X) = \deg R_i(X)$. \square

The other potential problem of using approximations to $R_i(Y)$ is that, if we do not use sufficient accuracy, the factorization might be changed. Corollaries 8.7 and 8.19 give bounds on the accuracy needed to preserve the correct factorization.

Corollary 8.7. Suppose that $R(Y)$, $B_0(Y)$, and $C_0(Y)$ are polynomials in Y of degrees k , l , and m , respectively, and

$$\|R(Y) - B_0(Y)C_0(Y)\| < |\text{Res}_Y(B_0(Y), C_0(Y))|^2. \quad (8.8)$$

Then if the polynomials $R(Y)$, $B_0(Y)$, and $C_0(Y)$ satisfy hypotheses 1, 2, and 3 of Hensel's Lemma, there exist an integer h and polynomials $u(Y)$ and $v(Y)$ such that h and the 5-tuple $(R(Y), B_0(Y), C_0(Y), u(Y), v(Y))$ satisfy the hypotheses and hence the conclusions of Hensel's Lemma.

Proof. Put

$$h = r \cdot \text{ord} \text{Res}(B_0(Y), C_0(Y)). \quad (8.9)$$

Then, using this value of h , hypothesis 4 of Hensel's Lemma is satisfied.

We will choose polynomials $u(Y)$ and $v(Y)$ in $K[Y]$ of degrees $\leq m - 1$ and $\leq l - 1$, respectively, to satisfy

$$u(Y)B_0(Y) + v(Y)C_0(Y) = 1. \quad (8.10)$$

Suppose that $u(Y) = \sum_{i=0}^{m-1} u_i Y^i$ and $v(Y) = \sum_{i=0}^{l-1} v_i Y^i$. Equation (8.10) amounts to a system of $l + m$ linear equations in the $l + m$ unknowns, u_0, u_1, \dots, u_{m-1} and v_0, v_1, \dots, v_{l-1} . The matrix of this system of linear equations is, up to sign, the Sylvester (resultant) matrix of $B_0(Y)$ and $C_0(Y)$ (see, for example, [10], Section 3.3.2). Since the determinant of this matrix is non-zero, the coefficients of $u(Y)$ and $v(Y)$ are uniquely determined elements of K , not all 0. We may estimate them as elements of the field \tilde{K} . Choose $\tau \in \tilde{K}$ to satisfy $\tau^r = \pi^{-s}$ so that $|\tau| = \|\pi^{-r/s}\| = 1/\lambda$. Put

$$\begin{aligned} u^\tau(Y) &= u(Y/\tau), & B_0^\tau(Y) &= B_0(Y/\tau), \\ v^\tau(Y) &= v(Y/\tau), & C_0^\tau(Y) &= C_0(Y/\tau). \end{aligned} \quad (8.11)$$

Then,

$$\begin{aligned} \|u^\tau(Y)\|_1 &= \|u(Y)\|, & \|B_0^\tau(Y)\|_1 &= \|B_0(Y)\|, \\ \|v^\tau(Y)\|_1 &= \|v(Y)\|, & \|C_0^\tau(Y)\|_1 &= \|C_0(Y)\|. \end{aligned} \quad (8.12)$$

Substituting Y/τ for Y , equation (8.10) becomes

$$u^\tau(Y)B_0^\tau(Y) + v^\tau(Y)C_0^\tau(Y) = 1 \quad (8.13)$$

As above, equation (8.13) may be considered as a system of linear equations in the coefficients of $u^\tau(z)$ and $v^\tau(z)$, which may be obtained from the matrix of equation (8.10) by elementary row operations, giving

$$|u_i/\tau^i| \leq 1/|\text{Res}_Y(B_0(Y), C_0(Y))|. \quad (8.14)$$

It follows that

$$\|u(Y)\| \leq |\text{Res}_Y(B_0(Y), C_0(Y))|^{-1}, \quad (8.15)$$

and similarly

$$\|v(Y)\| \leq |\text{Res}_Y(B_0(Y), C_0(Y))|^{-1}. \quad (8.16)$$

Thus the remaining hypotheses of Hensel's Lemma hold. \square

This corollary shows that if $R(Y)$ is computed to accuracy given by (8.9), then any factorization found will be correct. To show that a proof of irreducibility is also not changed by small perturbations of $R(Y)$, we first need two easy lemmas.

Lemma 8.17. *Suppose that $B(Y)$ and $C(Y)$ are polynomials in $K[Y]$ whose product $A(Y)$ is pure. Then both $B(Y)$ and $C(Y)$ are pure. Furthermore, the Newton diagrams of the three polynomials $A(Y)$, $B(Y)$, and $C(Y)$ have the same slope.*

Proof. This follows by repeated applications of Theorem 3.1 and Lemma 3.2 of Chapter 6 of [6].

Lemma 8.18. *Suppose that $A(Y)$ and $B(Y)$ are polynomials in $K[Y]$ of the same degree k . Suppose further that $\|A(Y) - B(Y)\| < \|A(Y)\|$. Then, if $A(Y)$ is pure, so is $B(Y)$ and their Newton diagrams have the same slope.*

Proof. Put $\alpha = \|A(Y)\|$. Suppose that $A(Y) = \sum_{i=0}^k a_i Y^i$ and that $B(Y) = \sum_{i=0}^k b_i Y^i$. Then $|a_i| \leq \alpha \lambda^{-i}$ and $|a_i - b_i| < \alpha \lambda^{-i}$. It follows that $|b_i| \leq \alpha \lambda^{-i}$. Since $|a_0| = |\alpha|$ and $|a_k| = |\alpha| \lambda^{-k}$, we see that $|b_0| = |\alpha|$ and that $|b_k| = |\alpha| \lambda^{-k}$.

Corollary 8.19. *Suppose that $R(Y)$ is an irreducible polynomial of degree n satisfying $\|R(Y)\| = 1$, so that, in particular, $R(Y)$ is pure. Suppose that the Newton diagram of $R(Y)$ has slope $-s/r \leq 0$. If $R_0(Y)$ is a polynomial of degree n satisfying $\|R_0(Y)\| = 1$ and $\|R_0(Y) - R(Y)\| < \min(1, |\Delta_{R_0}|)^2$, then $R_0(Y)$ is irreducible.*

Proof. Suppose that $R_0(Y)$ factors as $R_0(Y) = B_0(Y)C_0(Y)$. By Lemma 8.18, $R_0(Y)$ is pure, and by Lemmas 8.17 both $B_0(Y)$ and $C_0(Y)$ are pure, and their Newton diagrams have slope $-s/r$. We may assume that $\|B_0(Y)\| = \|C_0(Y)\| = 1$. Using the definitions and standard properties of the resultant and discriminant (see Lang [19, pp 200–204]), we find that $|\Delta_{R_0}| = |\Delta_R|$, and that $|\text{Res}(B_0(Y), C_0(Y))| \geq |\Delta_R|$. Hence

$$\|R(Y) - R_0(Y)\| < |\text{Res}(B_0(Y), C_0(Y))|^2. \quad (8.20)$$

By Corollary 8.7, $R(Y)$ factors, contradicting the hypotheses. \square

Theorem 8.21. *Let K be an extension of degree k of \mathbb{Q}_p , and $F(X) \in K[X]$ have degree n . Algorithm **p-adic Factor** will find an irreducible factor of $F(X)$ in random time*

$$O(n^{8+\epsilon} \log^3 |\Delta_F| \log^2 p^k). \quad (8.22)$$

Proof. By Corollaries 8.7 and 8.19, we will find the correct factorization if we compute terms to $O(|\Delta_F|^2)$ precision. Note that, although we are starting in an extension of degree k of \mathbb{Q}_p , we may need to go to an extension of degree n of that field.

The dominant computation is the resultant, which in worst case takes time $O(n^4 \log^2(|\Delta_F|^2 np^{nk}))$ (see [10], Section 3.3). From the discussion in Section 6.2, the outer loop of the algorithm will be executed at most $O(\log n)$ times, and the inner loop at most $O(n^2 \log |\Delta_F|)$ times. When **Hensel Factor** succeeds, we may have to call **p-adic Factor** on a factor of degree at most $n/2$, so that no more than $O(\log n)$ recursive calls will be needed. Combining these bounds, we have (8.22). \square

The implied constant in (8.22) depends upon the choice of uniformizer π and representatives \mathcal{A} . Note that this is a pessimistic worst-case bound. Most polynomials factor on the first call to **Hensel Factor**, and it takes an effort to construct a polynomial which goes through the inner and outer loops more than once. Since we have not used fast arithmetic algorithms, and it is unlikely that all the worst cases can occur simultaneously, with a more detailed analysis the $n^{8+\epsilon}$ in (8.22) can be improved.

References

- Emil Artin. *Algebraic numbers and algebraic functions*. Gordon and Breach, 1967.
- C. Batut, K. Belabas, D. Bernardi, H. Cohen, and M. Olivier. User's guide to PARI-GP, for version 2.0.10, July 1998.
- Elwyn R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24:713–735, 1970.
- W. Bosma, J. Cannon, and C. Playoust. The magma algebra system I: The user language. *J. Symb. Comp.*, 24:235–269, 1997.
- David G. Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36:587–592, 1981.

6. J. W. S. Cassels. *Local Fields*. Cambridge University Press, 1986. ISBN 0-521-30484-9 (hard cover) or 0-521-31525-5 (paper back).
7. A. L. Chistov. Efficient factorization of polynomials over local fields. *Soviet Math. Doklady*, 35:430–433, 1987. Translated from Russian original.
8. A. L. Chistov. Efficient factoring polynomials over local fields and its applicatons. In *Proceedings of the international congress of mathematicians, Kyoto, Japan, August 21-29, 1990*, pages 1509–1519, Vol. 2. Springer Verlag, 1991. ISBN 0-387-70047-1.
9. A. L. Chistov. Algorithm of polynomial complexity for factoring polynomials over local fields. *Journal of mathematical sciences*, 70:1912–1933, 1994. Translated from Russian original.
10. Henri Cohen. *A course in computational algebraic number theory*. Springer Verlag, 1994. ISBN 0-387-55640-0 or 3-546-55640-0.
11. David Ford and Pascal Letard. Implementing the round four maximal order algorithm. *Journal de Théorie des Nombres des Bordeaux*, 6:33–80, 1994.
12. Patrizia Gianni, Victor Miller, and Barry Trager. Decomposiition of algebras. Unpublished.
13. Daniel M. Gordon. <http://sdcc12.ucsd.edu/~xm3dg>. Web Site.
14. Fernando Gouvea. *p-adic Numbers*. Springer-Verlag, 1993. ISBN 0-387-56844-1.
15. W. B. Gragg. The Padé table and its relation to certain algorithms of numerical analysis. *SIAM Review*, 14:–62, 1972.
16. Helmut Hasse. *Number Theory*. Springer Verlag, 1980. ISBN 0-387-08275-1.
17. Kurt Hensel. *Theorie der Algebraischen Zahlen*. B. G. Teubner, 1908.
18. Dexter Kozen. Efficient resolution of singularities of plane curves. In *Proceedings 14th conference on foundations of software technology and theoretical computer science*, 1994.
19. Serge Lang. *Algebra*. Addison-Wesley, 1984.
20. R. Loos. Generalized polynomial remainder sequences. In B. Buchberger, G. E. Collins, and R. Loos, editors, *Computer Algebra Symbolic and Algebraic Computation, second edition*, pages 115–136. Springer Verlag, 1983. ISBN 0-387-81776-X.
21. Daniel A. Marcus. *Number Fields*. Springer-Verlag, 1977. ISBN 0-387-90279-1 or 3-540-90279-1.
22. Władysław Narkiewicz. *Elementary and analytic theory of algebraic numbers, second edition*. Springer Verlag, 1989. ISBN 0-387-51250-9.
23. Michael E. Pohst. *Computational algebraic number theory*. Birkhäuser Verlag, 1993. ISBN 0-8176-2913-0 or 3-7643-2913-0.
24. Michael E. Pohst and Hans Zassenhaus. *Algorithmic algebraic number theory*. Cambridge University Press, 1989. ISBN 0-521-33060-2.
25. Paulo Ribenboim. *The theory of classical valuations*. Springer Verlag, 1999. ISBN 0-387-98525-5.
26. Ian Stewart and David Tall. *Algebraic Number Theory*. Chapman and Hall, 1987. ISBN 0-412-29870-8 or 0-412-29690-X.
27. André Weil. *Basic Number Theory*. Springer Verlag, 1970.
28. Edwin Weiss. *Algebraic Number Theory*. McGraw-Hill, 1963.
29. Richard Zippel. *Effective Polynomial Computation*. Kluwer Academic Press, 1993. ISBN 0-7923-9375-9.

Strategies in Filtering in the Number Field Sieve

Stefania Cavallar

Centrum voor Wiskunde en Informatica
Postbus 94079, 1090 GB Amsterdam, The Netherlands
`Stefania.Cavallar@cwi.nl`

Abstract. A critical step when factoring large integers by the Number Field Sieve [8] consists of finding dependencies in a huge sparse matrix over the field \mathbb{F}_2 , using a Block Lanczos algorithm. Both size and weight (the number of non-zero elements) of the matrix critically affect the running time of Block Lanczos. In order to keep size and weight small the relations coming out of the siever do not flow directly into the matrix, but are filtered first in order to reduce the matrix size. This paper discusses several possible filter strategies and their use in the recent record factorizations of RSA-140, R211 and RSA-155.

Introduction

The Number Field Sieve (NFS) is the asymptotically fastest algorithm known for factoring large integers. It holds the records in factoring special numbers (R211 [4]) as well as general numbers (RSA-140 [3] and RSA-155 [5]). One disadvantage is that it produces considerably larger matrices than other methods, such as the Quadratic Sieve [1]. Therefore it is more and more important to find ways to limit the matrix size. This can be achieved by using good sieving parameters and by “intelligent” filtering.

In this paper we describe the extended version of the program `filter` which we implemented following ideas of Peter L. Montgomery. Its goal is to speed up Block Lanczos’s running time by reducing the matrix size but still keeping the weight under control.

A previous implementation of the program `filter` [8, section 7] did 2- and 3-way merges. When using Block Lanczos, higher-way merges were commonly banned from the filter step in order to limit the matrix weight. For instance, also James Cowie et al. [6, section Cycles] explicitly avoided merges higher than 3 for the factorization of RSA-130.

The most important new ingredients of the present `filter` implementation are an algorithm to discard excess relations and “controlled” higher-way merges. We determine arithmetically which merges reduce Block Lanczos’s running time.

For the factorization of RSA-140 only 2- and 3-way merges were performed which led to a matrix of 4.7 million columns. With the present filter strategy we could have saved up to 33% of linear algebra time by reducing the size to 3.3 million columns. For the factorization of R211 we already used an intermediate `filter` version which did 4- and 5-way merges, but we could still get an improved

matrix after the factorization. For RSA-155, we could take full advantage of the present version and did “controlled” merges up to prime ideal frequency 8 which led to a matrix of 6.7 million columns and an average of 62 entries per column which was used to factor the number. Afterwards, we were able to reduce this size to 6.3 million columns.

First, we give a brief description of the NFS. Secondly, the `filter` implementation will be described with special focus on the new features. In section 3 we will describe other filter strategies we came across in the literature and compare it with our approach. Finally, experimental results for RSA-140, R211 and RSA-155 are listed and interpreted.

1 Brief Description of NFS

We briefly describe the NFS factoring method here, skipping parts which are not relevant for the understanding of this paper such as the sieving step itself.

By N we denote the composite number we would like to factor. We select an integer M and two irreducible polynomials $f(x)$ and $g(x) \in \mathbb{Z}[x]$ with $\text{cont}(f) = \text{cont}(g) = 1$ and $f \neq \pm g$ such that $f(M) \equiv g(M) \equiv 0 \pmod{N}$. By $\alpha, \beta \in \mathbb{C}$ we denote roots of $f(x)$ and $g(x)$, respectively.

The goal is to construct a non-empty set S of co-prime integer pairs (a, b) for which both $\prod_{(a,b) \in S} (a - b\alpha)$ and $\prod_{(a,b) \in S} (a - b\beta)$ are squares, say, $\gamma^2 \in \mathbb{Z}[\alpha]$ and $\delta^2 \in \mathbb{Z}[\beta]$, respectively. Once we have found S , the two natural ring homomorphisms $\phi_1 : \mathbb{Z}[\alpha] \rightarrow \mathbb{Z}/N\mathbb{Z}$ mapping α to M and $\phi_2 : \mathbb{Z}[\beta] \rightarrow \mathbb{Z}/N\mathbb{Z}$ mapping β to M as well, yield the congruence

$$\phi_1(\gamma)^2 \equiv \phi_1(\gamma^2) \equiv \prod_{(a,b) \in S} (a - bM) \equiv \phi_2(\delta^2) \equiv \phi_2(\delta)^2 \pmod{N}.$$

which has the desired form $X^2 \equiv Y^2 \pmod{N}$. By computing $\gcd(X - Y, N)$ we may find a divisor of N . The major obstruction in this series of congruences is that we need to find $\gamma \in \mathbb{Q}(\alpha)$ from γ^2 (and δ from δ^2 , respectively). See Montgomery’s [15] or Phong Nguyen’s [17] papers for a description of their square root algorithms.

How to find the set S ? We write

$$F(x, y) = f(x/y)y^{\deg(f)} \quad \text{and} \quad G(x, y) = g(x/y)y^{\deg(g)}$$

for the homogeneous form of $f(x)$ and $g(x)$, respectively. Consider $a - b\alpha \in \mathbb{Q}(\alpha)$ and $a - b\beta \in \mathbb{Q}(\beta)$. The minus sign is chosen in order to have

$$N_{\mathbb{Q}(\alpha)/\mathbb{Q}}(a - b\alpha) = F(a, b)/c_1 \quad \text{and} \quad N_{\mathbb{Q}(\beta)/\mathbb{Q}}(a - b\beta) = G(a, b)/c_2,$$

where the c_i ’s are the respective leading coefficients of $f(x)$ and $g(x)$.

After the sieving we are left with many pairs (a, b) such that $\gcd(a, b) = 1$ and both $F(a, b)$ and $G(a, b)$ are products of primes smaller than the large prime

bounds L_1 and L_2 , respectively, which were chosen by the user before the sieving. The pairs (a, b) are commonly denoted as *relations*. A necessary condition for

$$\prod_{(a,b) \in S} (a - b\alpha) \quad \text{and} \quad \prod_{(a,b) \in S} (a - b\beta)$$

to be squares is that the norms

$$N_{\mathbb{Q}(\alpha)/\mathbb{Q}} \left(\prod_{(a,b) \in S} (a - b\alpha) \right) \quad \text{and} \quad N_{\mathbb{Q}(\beta)/\mathbb{Q}} \left(\prod_{(a,b) \in S} (a - b\beta) \right)$$

are squares. Therefore we require S to have even cardinality and

$$\prod_{(a,b) \in S} F(a, b) \quad \text{and} \quad \prod_{(a,b) \in S} G(a, b)$$

to be squares. The condition is not sufficient because elements having the same norm may differ from each other (not only by units!). Let p be a prime divisor of $F(a, b) = f(a/b)b^{\deg(f)}$. We distinguish two cases:

- $p \mid f(a/b)$. This means that $a/b \equiv q \pmod{p}$ with $0 \leq q < p$ is a root of $f(x)$ modulo p . In the sequel such a p is referred to as p, q .
- $p \nmid b$. Since $\gcd(a, b) = 1$ it follows that $p \nmid a$ and therefore $p \mid c_1$. This can happen for a small set of primes only, since the leading coefficient is of limited size. These roots are called *projective roots* and denoted as p, ∞ .

We will call the couples p, q , where q is allowed to be ∞ , *prime ideals*, since they are in bijective correspondence with the first degree prime ideals of the ring $\mathbb{Z}[\alpha] \cap \mathbb{Z}[\alpha^{-1}]$. See [2, Section 12.6].

Consequently, we write

$$|F(a, b)| = \prod_{p,q} p^{e_1(a, b, p, q)} \quad \text{and} \quad |G(a, b)| = \prod_{p,q} p^{e_2(a, b, p, q)}.$$

In order for $\prod_{(a,b) \in S} F(a, b)$ and $\prod_{(a,b) \in S} G(a, b)$ to be squares in $\mathbb{Q}(\alpha)$ and $\mathbb{Q}(\beta)$, respectively, we require all the exponents in

$$\prod_{(a,b) \in S} |F(a, b)| = \prod_{p,q} p^{\sum_S e_1(a, b, p, q)} \quad \text{and} \quad \prod_{(a,b) \in S} |G(a, b)| = \prod_{p,q} p^{\sum_S e_2(a, b, p, q)}$$

to be even. This condition can be stated in terms of the field \mathbb{F}_2 as well. We just think of a relation (a, b) as a vector in \mathbb{F}_2 whose first entry is 1 (in order to control the parity of S) and the following entries are given by the exponents $e_1(a, b, p, r)$ and $e_2(a, b, p, r)$ modulo 2. A 1 signals the occurrence of an uneven power of a prime ideal. The task of finding some suitable sets S translates now into finding dependencies modulo 2 between the columns of a matrix which is built up with the relation vectors given by the siever. We need to have enough relations to guarantee that the matrix provides enough dependencies.

Alas, not every dependency yields a set S such that $\prod_{(a,b) \in S} (a - b\alpha)$ and $\prod_{(a,b) \in S} (a - b\beta)$ are squares, but we can make the method practical by producing several dependencies and doing quadratic character tests [2, Section 8].

The filter stage occurs between the sieving step and the linear algebra step of the NFS. It is a preliminary linear algebra process since it corresponds to dropping columns (*pruning*) and adding up columns modulo 2 (*merging*).

2 Description of the New Filter Tasks

We distinguish 19 merge levels: level 0 and 1 fall into pruning, level 2 through 18 within merging.

We shall say that a prime ideal p, q is *(un)balanced* in a relation (a, b) if it appears to an (un)even number in $F(a, b)$ or $G(a, b)$ ^{*}. We distinguish between prime ideals of norm below and above a user determined bound `filtmin`. Accordingly, we speak about *small* and *large* prime ideals. We will denote prime ideals p, q by I . We write a relation $r = r(a, b)$ as the collection of its unbalanced large prime ideals, $r : I_1, I_2, \dots, I_k$. Merging means combining relations which have a common prime ideal in order to balance it. For example, if I appears only in $r_1 : I_{10} = I, I_{11}, \dots, I_{1k_1}$ and $r_2 : I_{20} = I, I_{21}, \dots, I_{2k_2}$, we can combine the two relations into $r_1 + r_2 : I_{11}, \dots, I_{1k_1}, I_{21}, \dots, I_{2k_2}$ with the result that I is balanced in $r_1 + r_2$. More generally, a *k-way merge* is the procedure of combining k relations with a common prime ideal I into $k - 1$ relation pairs without I . By a *relation-set* we mean a single relation, or a collection of two or more relations generated by a merge. We do merges up to prime ideal frequency 18. The parameter `mergelevel` l means that k -way merges with $k \leq l$ may be performed. The weight of a relation-set r , i.e., the number of unbalanced prime ideals in it, is denoted by $w(r)$.

2.1 Pruning

As the verb “pruning” suggests, this part of the program removes unnecessary relations from the given data, that is *duplicates* and *singletons* and, if the user wants to, also excess relations. Duplicates are obviously superfluous and singletons cannot be part of a winning set S since they contain a prime ideal which does not occur in any other relation and can subsequently not be combined to form a square. If the difference between the number of relations and the number of large prime ideals outnumbers a user-chosen bound (`keep`), the *clique* algorithm selects relations to delete.

`mergelevel` 0 only removes duplicates and can be used to merge several sieving outputs to a single file, possibly before sieving completes. `mergelevel` 1

* In very rare cases (p divides the polynomial resultant) we can have the same p, q appearing in both F and G . Recall that they are *not* the same, since they correspond to ideals in different rings. We abstain from labeling the ideals accordingly, for the sake of simplicity.

will only be performed if the full set of relations is available and covers algorithms for the removal of duplicates, singletons and excess relations.

Duplicates. First we want to eliminate duplicate relations. They may arise for various reasons. Most commonly they come from sieving jobs that were stopped and later restarted. In case of a *line-by-line siever* [8, section 6] the resumed jobs start with the last b sieved by the previous job; this is the only way that duplicates arise. In case of a *lattice siever* [18] the job starts with the special prime ideal I sieved last, and will generate duplicates, or it can do so because a relation may contain, apart from its own special I , other prime ideals that are used as special prime ideals as well. The simultaneous use of line-by-line and lattice siever also causes overlap.

Duplicates are tracked down by hashing [12]. Since it is easier and cheaper to use a number instead of a relation as a hash table entry, we “identify” a relation with a number. The user specifies how many relations he expects to be in the input file(s) (`maxrelsinp`). This figure is used to choose the size of the in-memory tables needed during the pruning algorithm. The program reads in relation after relation. In order to detect duplicates, the program maps each relation (a, b) to an integer between 0 and $2^{64} - 1$. The mapping function, $h = h(a, b)$, should be nearly injective since relations mapped to the same value will be treated as duplicates. It is rather easy to construct such a function, since even a huge amount of relations, say 200 million (for RSA-155 we had to handle 124.7 million relations), is small compared to the 2^{64} possible function values. With 64 bits for the function value we expect about

$$\frac{\binom{2 \cdot 10^8}{2}}{2^{64}} \approx 0.0011$$

false duplicates, which means that there will hardly be any false duplicates. With 32 bits only, this number would amount to about $4.7 \cdot 10^6$, which is a fair proportion of all relations.

The function $h(a, b)$ is defined as follows. It takes values of a and b up to 2^{53} . Put $\Pi = \lfloor \pi \cdot 10^{17} \rfloor$ and $E = \lfloor e \cdot 10^{17} \rfloor$. We have $\gcd(\Pi, E) = 1$. Define

$$H(a, b) = \Pi a + Eb.$$

If $H(a_1, b_1) = H(a_2, b_2)$ and $(a_1, b_1) \neq (a_2, b_2)$ we have

$$\frac{a_1 - a_2}{b_1 - b_2} = -\frac{E}{\Pi}$$

which is impossible, since $|a|$ and $|b|$ are known to be much smaller than $\Pi/2$ and $E/2$, and $\gcd(\Pi, E) = 1$. Define $h(a, b) = H(a, b) \bmod 2^{64}$. Since H is injective, false duplicates for h can only come from the truncation modulo 2^{64} .

The function values of h again are mapped by a hash function into a hash table. If the user has specified `mergelevel 0`, the non-duplicates are written to

the output file whereas, if the user has chosen `mergelevel 1`, the non-duplicate relations are memorized in a table for further processing, while considering only the large prime ideals. In the sequel, we shall call this table the *relation table*.

Singletons. If both polynomials f and g split completely into distinct linear factors modulo a prime p which does not divide the leading coefficients, we get a so-called free relation corresponding to the prime ideal factorization of the elements $p = p - 0\alpha$ and $p = p - 0\beta$ of norm $N_{\mathbb{Q}(\alpha)/\mathbb{Q}}(p) = F(0, p)/c_1 = p^{\deg(f)}$ and $N_{\mathbb{Q}(\beta)/\mathbb{Q}}(p) = G(0, p)/c_2 = p^{\deg(g)}$, respectively. Approximately $1/(g_f \cdot g_g)$ of the primes offer a free relation, where g_f and g_g are the orders of the Galois groups of the polynomials f and g , respectively [10]. The free relation $(p, 0)$ is added to the relation table only if all prime ideals of norm p appear in the relation table.

Next, a frequency table is built for all occurring prime ideals which is adjusted as the relation table changes. The relation table is then scanned circularly and relations containing an ideal of frequency 1 (singletons) are removed from it. The program executes as many passes through the table as is needed to remove all singletons.

At the end of the pruning algorithm we would like the remaining number of relations to be larger than the total number of prime ideals. Therefore we need to reserve a surplus of relations for the small prime ideals: Per polynomial, the number of prime ideals below `filtmin` is approximately $\pi(\text{filtmin})$, i.e., the number of primes below `filtmin`, see [14]. Consequently, we require a surplus of approximately $(2 - (g_f \cdot g_g)^{-1}) \cdot \pi(\text{filtmin})$ relations. If the required surplus is not reached we need to sieve more relations.

Clique Algorithm. If there are sufficiently many more relations than ideals, the user may want to specify how many more relations than large ideals to retain after the pruning stage (`keep`).

In [19, step 3] Pomerance and Smith eject excess relations by simply deleting the heaviest relations. However, as an alternative, they suggest to delete relations which contain many primes of frequency 2. Our approach is similar to this alternative. The algorithm we use is called *clique algorithm*, since it deletes relations that stick together.

Consider the graph with the relations from the relation table as nodes. We connect two nodes if the corresponding relations would be merged in a 2-way merge. The components of the graph are called cliques. The relations in a clique are close to each other in the sense that if one of them is removed, the others will become singletons after some steps and are therefore useless.

The clique algorithm determines all the cliques, evaluates them with the help of a metric and at each step keeps up to a prescribed number of them in a priority heap [12, page 144], ordered by the size of a metric value. The metric being used weighs the contribution from the small prime ideals by adding 1 for each relation in the clique and 0.5 for each free relation. The large prime ideals

which occur more than twice in the relation table contribute 0.5^{f-2} where f is the prime ideal's frequency. This way we “penalize” ideals with low frequency. Relation-sets containing many ideals with low frequencies are more likely to be deleted than those containing mainly high frequency ideals. By deleting these low-frequency relation-sets we hope to reduce especially low frequencies even more and get new merge candidates.

Finally, the relations belonging to cliques in the heap are deleted from the relation table. When deleting relations we decrease the ideal frequencies of the primes involved. Singletons may arise and we therefore continue with the singleton processing step. The clique algorithm may be repeated if the number of excess relations does not approximate `keep` sufficiently.

After duplication, singleton and possibly clique processing the relations are read again and only the non-free relations^{**} appearing in the relation table are written to the output file. If the input files have grown in the meantime, the new relations are discarded.

2.2 Merging

First, we have a closer look at how merging works, which parameters can be given and at how to minimize the weight increase during a k -way merge. Next, we give details about the implementation of the “controlled” merges. Finally we study the influence of merging on Block Lanczos's running time.

Merging aims at reducing the matrix size by combining relations. Throughout this section we give figures about weight changes in the matrix. These figures do not take account of possible other primes that may have been balanced incidentally during the same merge.

Parameters `mergelevel`, `maxpass`, `maxrels` and `maxdiscard`. With the parameter `mergelevel` the user specifies the highest k for which k -way merges are allowed to be executed. The user fixes the maximum number (`maxpass`) of *shrinkage passes* to execute. During a shrinkage pass, all large primes are checked once and possibly merged, see [8, section 7] for more details.

The simplest case is the so-called 2-way merge. A prime ideal I is unbalanced in exactly two relations, r_1 and r_2 , and we combine the relations into the relation-set $r_1 + r_2$. As a result, we have one fewer column (r_1 and r_2 disappear, $r_1 + r_2$ enters) as well as one fewer row (prime ideal I) and the total weight has thereby decreased by 2.

In general, if a prime ideal I is unbalanced in exactly k relations ($k \geq 2$)***, we can choose $k - 1$ independent relation pairs out of the possible $\binom{k}{2}$ pairs. For example, if $k = 3$, there are 3 possible ways to combine the 3 relations involved, r_1 , r_2 and r_3 , to a couple, namely $r_1 + r_2$, $r_2 + r_3$ and $r_1 + r_3$. Each one can be

^{**} Free relations will be generated during the merge stage again.

^{***} The case $k = 1$ denotes a singleton which would be deleted.

obtained from the other two, for instance $r_1 + r_3 = (r_1 + r_2) + (r_2 + r_3)$ as all the prime ideals of r_2 are balanced since r_2 appears twice.

After the merge, the prime ideal I is balanced. Its corresponding row has disappeared from the matrix. The total gain of every merge consists in fact in one fewer column and one fewer row. The drawback of merging is, of course, matrix fill-in. A 2-way merge causes no fill-in at all, we even have 2 entries fewer in the matrix. However, a k -way merge, $k \geq 3$, causes the matrix to be heavier by about the weight of $k - 2$ relations minus the $2(k - 1)$ entries that disappeared.

If the matrix is going to be “lopsided”, i.e., if it has many more relations than ideals, it is useful to drop heavy relation-sets. The program therefore discards the ones which contain more relations than the user-determined bound `maxrels`.[†] The user may specify `maxdiscard`, that is, the maximum number of relation-sets to be dropped during one `filter` run. Once `maxdiscard` has been reached, k -way merges, $k \geq 3$, are inhibited.

Minimizing the Weight Increase of a k -Way Merge. Which $k - 1$ of the possible $\binom{k}{2}$ relation pairs should be chosen in order to achieve the lowest weight increase? First of all, each relation has to appear in at least one relation couple, that is, we need to form independent relation sets, in order not to loose data. Secondly, we focus on minimizing the weight increase. In the beginning, when all relations are true single relations, we usually achieve the lowest weight increase by choosing the lightest relation (*pivot*) and combining it with the remaining $k - 1$ relations. We call this *pivoting*. More precisely, this happens always when no additional prime ideals except for the prime ideal I become balanced in any of the candidate relation couples. If we assume the pivot relation to be r_k , the weight increase Δw will be exactly

$$\Delta w = (k - 2)w(r_k) - 2(k - 1). \quad (1)$$

The choice becomes more complicated, when additional prime ideals get balanced, especially when we are merging already combined relation-sets. For example, consider the following 5 relations, which are candidates for two 3-way merges with the prime ideals I and J :

- $r_1 : I$ and $v - 1$ other prime ideals
- $r_2 : I$ and $v - 1$ other prime ideals
- $r_3 : I, J$ and $v - 2$ other prime ideals
- $r_4 : J$ and $v - 1$ other prime ideals
- $r_5 : J$ and $v - 1$ other prime ideals

For the sake of simplicity, we assume that all the relations have the same weight v and do not share other primes except for I and J . Imagine, r_3 is used as a

[†] We weigh a free relation less than 1 (we used 0.5), because, even if it may have several large primes, it should have less total weight.

pivot relation to eliminate J . We get

$$\begin{aligned} r_1 + r_3 &: J \text{ and } 2v - 3 \text{ other prime ideals} \\ r_2 + r_3 &: J \text{ and } 2v - 3 \text{ other prime ideals} \\ r_4 &: J \text{ and } v - 1 \text{ other prime ideals} \\ r_5 &: J \text{ and } v - 1 \text{ other prime ideals} \end{aligned}$$

Now J appears 4 times, so we need a 4-way merge to balance it. For the elimination of J the two relations r_4 and r_5 seem the best pivot candidates in a 4-way merge, since they have lowest weight. However, pivoting with r_5 results into

$$\begin{aligned} (r_1 + r_3) + r_5 &: 3v - 4 \text{ prime ideals} \\ (r_2 + r_3) + r_5 &: 3v - 4 \text{ prime ideals} \\ r_4 + r_5 &: 2v - 2 \text{ prime ideals} \end{aligned}$$

with total weight $8v - 10$, whereas

$$\begin{aligned} (r_1 + r_3) + (r_2 + r_3) &: 2v - 2 \text{ prime ideals} \\ (r_1 + r_3) + r_5 &: 3v - 4 \text{ prime ideals} \\ r_4 + r_5 &: 2v - 2 \text{ prime ideals} \end{aligned}$$

ends with weight $7v - 8$ [‡]. When $v > 2$ we have $8v - 10 > 7v - 8$ which indicates that we should not stick to pivoting for all the merges.

The problem of minimizing the weight increase can be stated using graphs. The vertices are given by the k relations which are candidates for a k -way merge and the $\binom{k}{2}$ edges between them represent possible merges. The edge between two nodes r_i and r_j has weight $w(r_i + r_j)$. Given this weighted graph we wish to select a tree with minimum total weight. The solution is called a *minimum spanning tree* [11, page 460]. This problem is a well-known problem of combinatorial optimization. In order to solve it we use the algorithm as formulated by Jarník [9, pages 46–47].

Implementation of “Controlled” Merges. We limit the weight increase of a single merge by requiring that a merge should not add more than a prescribed number, m_{max} , of original relations to the matrix. We give all the initial relations the same weight (except for free relations that weigh one half), which is reasonable since the relations are the factorizations of numbers of about the same size.

Let us consider k relation-sets which are candidates for a k -way merge. The individual relation-sets may contain several original relations. Suppose the lightest candidate relation-set has j relations, where free relations count for 0.5. Let c be the number of relation-sets with exactly this minimum number j of relations. Shrinkage pass 1 starts with $m = 1$ and we subsequently augment m up until m_{max} and allow for the k -way merge when $(k - 2)j \leq m - (c - 1)/2$. The

[‡] The latter situation is also achieved when first using r_1 as a pivot and then doing a 3-way merge with pivot relation r_5 .

m gives the maximum weight increase (in number of relations) allowed during a merge. We introduced c in order to postpone some merges and do the ones where the best way to merge is clear cut first. Since we are still interested in doing lower weight merges before higher weight merges we increase m only every other shrinkage pass and set $c = 1$ during these shrinkage passes. In most of the runs we had $m_{max} = 7$, but we tried $m_{max} = 8$ as well. Solving the inequality $(k-2)j \leq m_{max}$ for k gives $k \leq \frac{m_{max}}{j} + 2$. It follows that, with $m_{max} = 7$, merges with ordinary relations ($j = 1$) are limited to prime ideal frequency 9 whereas free relations ($j = 0.5$) can be used in merges up to prime ideal frequency 16. For the factorization of RSA-155 we performed merges up to prime ideal frequency 8.

Table 1 shows the maximum number of relations a pivot relation-set may consist of, for $m_{max} = 7$ and 8. Even if we are not pivoting, we ask at least one relation not to contain more relations than this bound.

k	$\left\lfloor \frac{m_{max}}{k-2} \right\rfloor / 2$	
	$m_{max} = 7$	$m_{max} = 8$
3	7	8
4	3.5	4
5	2	2.5
6	1.5	2
7	1	1.5
8–9	1	1
10	0.5	1
11–16	0.5	0.5
17–18	—	0.5

Table 1. Allowed number of relations in pivot relation-set for k -way merge

Influence of Merging on Block Lanczos’s Running Time. Given an $m \times n$ matrix, $n > m$, of total weight w , the running time estimate of Block Lanczos is given by $\mathcal{O}(wn) + \mathcal{O}(n^2)$ [16]. Both terms grow with n , so we will focus on reducing n . If we manage to reduce n by a certain factor while w does not grow by more than this factor, we will get a running time reduction, independently of the constants in the two terms. Moreover, we predict the constant in the $\mathcal{O}(n^2)$ term to be the larger one. Therefore, it is natural to write the running time as

$$\mathcal{O}((w + Cn)n) \tag{2}$$

with $C \geq 1$. Since we do not need absolute running times, we drop the \mathcal{O} -sign and use the function $t(n, w) = (w + Cn)n$. The larger the constant C , the more it will be convenient to reduce the matrix size. The constant depends on the implementation, for example on the number of bits per vector element (K) used[§].

[§] Montgomery [16] gives the formula $\mathcal{O}(wn/K) + \mathcal{O}(n^2)$ for the running time.

Montgomery (personal communication) at first estimated the constant C to be about 50. For some approximate values of C see Table 7 or Table 2.

Let us determine a bound for the weight increase Δw such that a merge causing an increase below this bound still is beneficial to the running time. The condition for Δw becomes

$$t(n-1, w + \Delta w) - t(n, w) < 0. \quad (3)$$

Inequality (3) is equivalent to

$$0 > n((1-2n)C - w + (n-1)\Delta w) = (n-1)(-2Cn - w + n\Delta w) - w - Cn.$$

The inequality is satisfied if $\Delta w < 2C + \frac{w}{n}$. It follows that the allowed weight increase grows with C and the average column weight $\frac{w}{n}$. That means that denser matrices allow heavier merges than sparser matrices do.

Let us calculate a limit for the pivot relation weight j of a general k -way merge, $k \geq 3$. According to equation (1) we require

$$\Delta w = (k-2)j - 2(k-1) < 2C + \frac{w}{n}.$$

which results into

$$j < \frac{2C + \frac{w}{n} + 2(k-1)}{k-2}. \quad (4)$$

In Table 2 we report the allowed pivot relation weights for merges up to prime ideal frequency 10. We chose $\frac{w}{n} = 30$ (typical after applying only 2- and 3-way merges) and $\frac{w}{n} = 50$ (typical $\frac{w}{n}$ of many of our final matrices). The horizontal lines divide between above and below $\frac{w}{n}$.

k	$2C + \frac{w}{n} + 2(k-1)$								$\frac{w}{n} = 50$
	$k-2$				-1				
$\frac{w}{n} = 30$	$C=49$	$C=37$	$C=14$	$C=1$	$C=49$	$C=37$	$C=14$	$C=1$	
	3	131	107	61	35	151	127	81	55
4	66	54	31	18	76	64	41	28	
5	45	37	21	13	51	43	28	19	
6	34	28	16	10	39	33	21	15	
7	27	23	13	8	31	27	17	12	
8	23	19	11	7	26	22	15	10	
9	20	17	10	6	23	19	13	9	
10	18	15	9	6	20	17	11	8	

Table 2. Allowed pivot relation weights for k -way merge

From Table 2 we can see that 3-way merges can be done with rather heavy pivot relations; even for $C = 1$ and $\frac{w}{n} = 50$ the allowed weight exceeds $\frac{w}{n}$. Denser matrices allow also for denser pivot relations.

By substituting $\frac{w}{n}$ for j in (4) we can derive a condition for when to do k -way merges for $k > 3$ with an average weighing pivot relation:

$$\frac{w}{n} < \frac{2C + 2(k - 1)}{k - 3} \quad (5)$$

The analysis for $k = 3$ has to be done separately, we require (3) for $\Delta w = \frac{w}{n} - 4$. By reorganizing the terms we get $-4(n - 1) - \frac{w}{n} - C(2n - 1) < 0$ which is always satisfied. This means that 3-way merges with an average weight pivot relation are always profitable, independently from the density of the matrix or the constant C .

Table 3 gives the allowed average weights when merging with an average weight pivot relation. If we assume $C < 50$ and we apply the merges in ascending order of prime ideal frequency, 6-way merges with average weighing pivot relations will not be worthwhile because after the 5-way merges we have seen in practice $\frac{w}{n}$ to be around 50, which is higher than the maximum value of 35.

k	$\frac{2C + 2(k - 1)}{k - 3}$		- 1	
	$C = 49$	$C = 37$	$C = 14$	$C = 1$
4	103	79	33	7
5	52	40	17	4
6	35	27	12	3
7	27	21	9	3
8	22	17	8	3
9	18	14	7	2
10	16	13	6	2

Table 3. Allowed average weights for k -way merge

3 Other Methods in the Literature

We would like to mention two articles about similar filter strategies. These are “Solving Large Sparse Linear Systems Over Finite Fields” of LaMacchia and Odlyzko from 1990[13] and “Reduction of Huge, Sparse Matrices over Finite Fields Via Created Catastrophes” of Pomerance and Smith from 1992[19]. Their strategies are similar to each other but differ in some points. Both were designed to reduce the initial data to a substantially smaller matrix. This matrix was allowed to be fairly dense since it was going to be processed by Gaussian elimination afterwards. In contrast, the purpose of our method is to reduce the matrix size but still keep it sparse in order to take advantage of the Block Lanczos method. They were dealing with matrices of size up to 300K, we with matrices of size up to 7M. Each reflects the maximum size that could be handled at the time.

Both other methods executed their operations on the matrix itself whereas we dealt with the raw relations. We identified relations with columns in the final matrix whereas they identified relations with rows. Nevertheless, for an easier comparison, we will stick to identify relations with columns in the present description.

They operate only on part of the matrix (active rows) where no fill-in takes place. The operations must be memorized in order to be repeated on the complete matrix afterwards. LaMacchia and Odlyzko store the history in core, whereas Pomerance and Smith keep a history file.

We will distinguish between the pruning and merging step, as in the description of our method. The weight they look at is only the weight of the active primes at that moment.

The pruning step does differ from our approach only in how to delete excess relations. Duplicates and singletons are removed as soon as possible, as in our approach. Pomerance and Smith choose to remove the excess immediately, whereas LaMacchia and Odlyzko remove the excess just before the “collapse” or “catastrophe” during the merge step. Both decide to drop the heaviest relations, but Pomerance and Smith indicate that one might try other strategies (as we did).

In the beginning of the merge stage, a small number of rows (the heaviest, which correspond to small primes) are declared inactive. Merges are done by pivoting with columns that have only one 1 in the active part. There is no fixed limit for the prime ideal frequency up to which to merge. Once all possible merges have been done and there are still 1’s in the active part, more rows (again the heaviest) are declared inactive and the merge step is repeated. This is repeated until the active part collapses. This procedure leads to very heavy matrices. To overcome this, LaMacchia and Odlyzko for example, extend the inactive part considerably after it has reached a certain critical size. This way fewer merges can be executed and the fill-in is confined. Nevertheless, the matrices still have high column weights: the lightest example given by LaMacchia and Odlyzko has an average of 115 entries per column for a $6.0 \cdot 10^4$ columns matrix which is much denser than our densest matrix, the $6.3 \cdot 10^6$ columns matrix from Table 11 having an average 81 entries per column[¶].

Initially, for a sparse matrix, merges are done with very light columns, since the inactive part is small and cannot contain many 1’s. Further on, pivot relations can be very heavy: very probably, the single 1 in the increasingly smaller active part mostly represents a large prime and goes together with many small prime factors, since all polynomial values are about the same size (Pomerance and Smith try to overcome this by also allowing merges with pivot columns having two 1’s in the active part of the matrix.). Moreover, they do not make a distinction between “original” pivot relations and already merged ones, which can be substantially heavier.

[¶] The column weight 70 given in Table 11 corresponds to the matrix obtained when dropping the prime ideals of norm below 40.

In our merge procedure we also merge with already merged relations, but this happens in a controlled way. We limit the number of original relations which can be added during a single merge. We also minimize the fill-in per merge by using a minimum spanning tree algorithm instead of the simpler pivoting, see Section 2.2. But here we also have to say, that we cannot guarantee to always get the cheapest merge, because we count the contribution from the large prime ideals but only *estimate* the contribution from the small prime ideals.

In 1995, Thomas Denny proposed a Structured Gaussian elimination preliminary step for Block Lanczos [7]. He estimated $C = 1$ for his own Block Lanczos program. We therefore also included $C = 1$ in Tables 2 and 3.

4 Experimental Results

The experiments were done with two versions of our program **filter**. Both of them include pruning facilities.

The first version was capable of doing merges up to prime ideal frequency 5 and corresponded to the old program [8, section 7] if invoked with `mergelevel` 2 or 3. With the first version the user needed to specify when to start with the 4- and 5-way merges. For example, in the tables about filter runs (Tables 5, 8 and 10) the notation 4(x) in column `mergelevel` means that 4-way merges started x shrinkage passes after 3-way merges started. 5(x-y) means that 4-way merges started x shrinkage passes after 3-way merges did, and 5-way merges started y shrinkage passes later than 3-way merges.

The present **filter** version does not need this information any more. It can do merges up to prime ideal frequency 18. The merges are done in order of weight increase (measured in numbers of original relations). All runs except RSA-155's B6 had $m_{max} = 7$.

Table 4 gives an overview of all pruning activities in our experiments for RSA-140, R211 and RSA-155. All the figures are in units of a million. With prime ideals we mean prime ideals above 10M; we need to reserve an excess of 1.3M relations for the small prime ideals. The non-duplicate relation counts differ so much due to the use of different large prime bounds. Apparent errors are due to rounding values to units of one million.

The figures in Tables 5–11 are given in units of a million (M) or a thousand (K). We labeled the experiments with capital letters. All experiments with the same letter started with the same `mergelevel` 1 run.

In Tables 5, 8 and 10, columns 2–6 are input parameters. Column 7–10 are results: column “sets” gives the number of relation-sets remaining after the run, column “discarded” gives the total number of relation-sets which were discarded during the run. “excess” gives how many more relations than the approximate total number of ideals we retained. It indicates how many more relations we might still throw away in a further run. “not merged” gives the number of large prime ideals of frequency smaller or equal to `mergelevel` among the output relations. For the runs with the new version we also report the number of output

number being factored experiment	RSA-140		R211		RSA-155			
	A	B	A	B	A	B	C	D
raw relations (1)	65.7	68.5	57.6		130.8			
duplicates (2)	10.6	11.9	10.6		45.3			
non-duplicates (3)=(1)−(2)	55.1	56.6	47.0		85.5			
free relations (4)	0.1	0.1	0.8		0.2			
prime ideals (5)	54.2	54.7	49.5		78.8			
excess (6)=(3)+(4)−(5)	1.1	2.0	−1.7		6.9			
singletons (7)	28.5	28.2	26.5		32.5			
relations left (8)=(3)+(4)−(7)	26.8	28.5	21.3		53.2			
prime ideals left (9)	21.5	22.6	18.5		42.6			
excess (10)=(8)−(9)	5.2	6.0	2.8		10.6			
clique relations (11)	17.6	18.7	7.4	0	34.1	33.0	29.6	22.9
relations left (12)=(8)−(11)	9.2	9.8	13.9	21.3	19.1	20.2	23.6	30.3
prime ideals left (13)	7.8	8.1	12.2	18.5	17.4	18.2	20.6	25.3
excess (=keep) (14)=(12)−(13)	1.4	1.7	1.7	2.8	1.7	2.0	3.0	5.0

Table 4. summary of `mergelevel 0` and `1` runs

relation-sets made of one single relation since among those could be candidates for future high-way merges.

The Block Lanczos code typically finds almost K dependencies [16], where K is the number of bits per vector element. This enables us to drop the heaviest rows which leads to substantially lighter matrices^{||}. We dropped the rows corresponding to prime ideals of norm smaller than 50 for R211, whereas for RSA-140 and RSA-155, which have both exceptionally many small prime ideals, we omitted the prime ideals of norm smaller than 40**. In addition, the Block Lanczos code truncates every $m \times n$ matrix by default to $m \times (m + K + 100)$.

The tables featuring matrix data (Tables 6, 9 and 11) are made of two parts. In the first part we state the real size ($m \times n$), weight (w) and average column weight ($\frac{w}{n}$) of the matrices built. The numbers between two lines express the changes in size (number of columns) and weight from one matrix to the smaller one as percentages. Note that a $i\%$ decrease in matrix size makes the term wn shrink as long as the weight does not increase by more than $\frac{100i}{100-i}\%$ which is slightly larger than $i\%$. The second part shows the effective weight (w_{eff}) after truncating the matrix to size $m \times (m + K + 100)$, the effective average column weight ($\frac{w_{eff}}{m+K+100}$) and the Block Lanczos timings from a Cray C90 and a Silicon Graphics Origin 2000. The timings can vary substantially according to

^{||} In particular, all quadratic character rows are omitted. The pseudo-dependencies being found for this reduced matrix must be combined to real dependencies afterwards.

^{**} These figures match with the implementation for $K = 64$. For $K = 128$, we could even have dropped the prime ideals up to norm 180. The resulting lighter matrices would have led to shorter timings for that implementation. However, for simplicity, we used the same matrices for both the $K = 64$ and the $K = 128$ versions.

the load on the machines (other jobs interacting with ours): time differences of 20% are not unusual. Aiming at a fair comparison we tried to run the matrices at times with comparable load. In our tables, comparable timings are written in the same column. Only one Block Lanczos job per number was completely executed. All times in the tables are extrapolations: we did a short run, took the time of the fastest iteration and multiplied it by the number of iterations $(m + K + 100)/(K - 0.76)$, see [16].

RSA-140

This 140-digit number was factored on February 2, 1999. The experiment series A started with 65.7M raw relations, B with 68.5M from 5 different sites. We removed 1.4M and 1.6M duplicates, respectively, with `mergelevel 0` runs on each contributor's data. The experiments in Table 5 start with the remaining 64.3M respectively 66.9M relations having 54.2M and 54.7M large prime ideals, respectively. After the pruning step (with `filtmin= 10M`) we need an excess of $\frac{239}{120}\pi(10M) = 1.3M$ for the small prime ideals. For a summary of `mergelevel 0` and 1 runs, see Table 4.

In this paragraph we only describe experiment series A. The `mergelevel 1` run on the whole bunch of data removed another 9.2M duplicates and added 0.1M free relations for large primes. Note, that at this point the excess $64.3M - 54.2M - 9.2M + 0.1M = 1.1M^{\dagger\dagger}$ was less than the needed 1.3M. The excess was sufficient only after removing the singletons, when we were left with 26.8M relations having 21.5M large prime ideals. The clique algorithm removed a total of 17.6M relations to approximate the excess of $1.4M = 9.2M - 7.8M$.

The factorization was done using matrix A1.1 which took 100h on the Cray. Only 2- and 3-way merges were performed, because the code for higher than 3-way merges was not ready by then. For logistic reasons we had built the matrix before we received all the data.

With the complete data (experiment series B) the excess was enough from the beginning. Furthermore, a matrix constructed from this data by applying the same filter strategy as for A1.1 would have performed better than A1.1 as one can imagine when comparing A1.1.2.1 to B1.2: both did merges up to prime ideal frequency 5 and the latter is smaller in size *and* weight.

We also tried `mergelevel 8` (B2) with $m_{max} = 7$ which was introduced only just before the factorization of RSA-155. The program stopped with k -way merges, $k \geq 3$ at shrinkage pass 10 after having deleted 381K relations. This means that only merges with a maximum weight increase of 6 original relations had been done. Matrix B2 beats the `mergelevel 5` matrix of the same series (B1.2).

In Table 6 one can see from the percentages that each size reduction should have a favourable effect on Block Lanczos's running time which is confirmed by the time column.

^{$\dagger\dagger$} The apparent arithmetical error is due to rounding all numbers to units of a million.

These experiments confirm our idea of the advantage of higher-way merges. They show that collecting more data than necessary is recommendable. It does not become clear, however, how much excess data one should keep after the pruning step.

experiment	mergelevel	filtmin	maxdiscard	maxrels	maxpass	sets	discarded	excess	not merged
A	1	10M	keep	1.4M		9.2M	46 040K	90K	-
A1	2	10M	-	4.0	6	6.0M	54K	36K	59
A1.1	3	10M	unlim.	10.0	10	4.7M	3K	33K	0
A1.1.1	4(0)	10M	20K	10.0	10	4.2M	20K	13K	243K
A1.1.2	4(0)	10M	20K	12.0	10	4.0M	14K	20K	0
A1.1.3	4(0)	10M	20K	11.0	10	4.0M	20K	13K	48K
A1.1.2.1	5(0-0)	8M	17K	15.0	10	3.5M	17K	4K	0
B	1	10M	keep	1.7M		9.8M	46 906K	384K	-
B1	4(5)	10M	300K	8.0	12	4.3M	170K	208K	6K
B1.1	5(1-3)	10M	200K	11.5	10	3.6M	85K	128K	1K
B1.2	5(1-3)	10M	200K	10.5	10	3.4M	200K	14K	28K
B2	8	10M	375K	8.0	15	3.3M	383K	1K 909K/455K	

Table 5. RSA-140 filter runs

exp.	matrix size	%	weight	%	col.w.	w_{eff}	col.w.	Cray	SGI ^a
A1.1	4 671K \times 4 704K	-11	151.1M	+32	32.1	147.4M	31.5	75h	59d 24d
A1.1.1	4 180K \times 4 193K	-4	163.1M	+8	38.9	161.3M	38.6	65h	56d 22d
A1.1.3	3 999K \times 4 012K	-4	168.7M	+3	42.0	166.8M	41.7	63h	54d 21d
A1.1.2	3 960K \times 3 980K	-1	171.1M	+1	43.0	168.1M	42.4	62h	53d 20d
A1.1.2.1	3 504K \times 3 507K	-12	191.3M	+12	54.5	190.8M	54.4	56h	51d 18d
B1.2	3 380K \times 3 394K	-3	178.8M	+52	52.7	176.8M	52.3	51h	46d 16d
B2	3 285K \times 3 286K	-3	182.1M	+2	55.4	182.0M	55.4	50h	43d 15d

Table 6. RSA-140 matrices

^a The second column gives timings from the $K = 128$ implementation.

With each timing column, we fitted a surface $t = s_1 n^2 + s_2 nw$ to the points (n, w, t) . The fits were done by gnuplot's implementation of the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm. The quotient s_1/s_2 corresponds to the C from (2). Table 7 gives some possible values for C .

Block Lanczos implementation	s_1	s_2	C
vectorized Cray code with $K = 64$	1.84 ± 0.06	0.0499 ± 0.0014	37 ± 2
SGI code with $K = 64$	0.86 ± 0.14	0.060 ± 0.003	14 ± 3
improved SGI code with $K = 128^a$	0.69 ± 0.08	0.0140 ± 0.0018	49 ± 12

Table 7. C values for different Block Lanczos implementations

^a This version ‘under development’ by Montgomery is being optimized for cache usage rather than vectorization. It is being redesigned to allow parallelization, but we used only one processor.

$C = 14$ is much smaller than we had initially expected. According to Table 2, with $C = 14$ and assuming $\frac{w}{n} = 30$ we have that 4-way merges are convenient with pivot relations up to weight 31, which is slightly above average whereas 5-way merges should be done with lighter than average (max. 21 entries) pivot relations. When assuming $\frac{w}{n} = 50$ the maxima are higher but below average also for 4-way merges.

Why then did the matrices, which were constructed by more or less brutally doing all possible 3-, 4- and 5-way merges^{††}, perform better than we would expect from looking at the figures in Table 3 and 2? It seems most merges were able to find a pivot relation with much smaller weight than average. Furthermore, we must consider that the inequalities (4) and (5) do not take account of the weight and size reduction obtained by discarding relation-sets which are made of more than `maxrels` relations. Some benefit also comes from the minimum spanning tree algorithm.

With $C = 49$ and $\frac{w}{n} = 30$, even above average 6-way merges can be beneficial.

R211

The following two tables give data concerning filter experiments with the special 211-digit number R211:= $(10^{211} - 1)/9$, which is a so-called “repunit”, since all its digits are 1. It was factored on April 8, 1999. Five sites produced a total of 57.6M raw relations. 1.2M duplicates were removed during `mergelevel 0` runs on the individual data. The experiment series A and B both started with the remaining 56.4M relations having 49.5M prime ideals of norm above 10M. This means that we had 6.9M more relations than prime ideals which seemed to be enough since we needed to reserve $\frac{23}{12}\pi(10M) = 1.3M$ more relations accounting for the small prime ideals. Unfortunately, the `mergelevel 1` run on the complete data set revealed 9.4M duplicates. The remaining 47.0M relations plus 0.8M free relations were *less* than the number of prime ideals. However, we did not need to sieve further since we had an excess after removing the 26.5M singletons. The clique algorithm started hence with 21.3M relations having 18.6M prime ideals of norm larger than 10M, which is an excess of 2.8M. See Table 4.

^{††} For A1.1.2.1, all possible merges up to prime ideal frequency 5, for prime ideals of norm larger than 8M, had been performed.

Experiment series A gives the parameters and results of the `filter` runs that led to the matrix that was used to factor the number; it took 120 hours on the Cray. B shows a different approach, where we kept 1.1M more relations than for A after the pruning step, leaving more choice for merging.

experiment	mergelevel	filtmin	maxdiscard	maxrels	maxpass	sets	discarded	excess	not merged
A	1	10M	keep	1.7M		13.9M	33 839K	433K	-
A1	4(5)	20M	300K	6.0	10	6.8M	304K	124K	1 637K
A1.1	5(5-10)	20M	15K	12.0	15	5.6M	15K	109K	796K
A1.1.1 ^a	5(5-10)	8M	50K	15.0	15	4.9M	n.a.	63K	n.a.
B	1	10M	keep	2.8M		21.3M	26 488K	1 484K	-
B1	4(5)	20M	1 300K	6.0	10	6.7M	1 310K	206K	1 410K
B1.1	5(5-10)	20M	170K	12.0	15	4.8M	170K	11K	97K
B1.1.1	5(1-3)	8M	10K	18.0	10	4.6M	4K	8K	2
B2	8	10M	1 400K	9.0	15	4.7M	1 421K	30K	1 244K/925K
B3	8	10M	1 400K	10.0	15	4.5M	1 423K	64K	918K/777K

Table 8. R211 filter runs

^a This run was done with the flag `regroup`, which splits up existing relation-sets and does merges from scratch, which leads to different relation-sets.

Both `mergelevel` 4 runs can actually be considered `mergelevel` 3 runs, since the maximum number of discards, `maxdiscard`, was reached before 4-way merges would have started.

exp.	matrix size	%	weight	%	col.w.	w_{eff}	col.w.	Cray	SGI
A1.1.1	4 820K × 4 896K	-0	234.2M	-5	47.8	221.2M	45.88	118h - 97h 93h	96d
B1.1	4 863K × 4 877K	-3	223.3M	+4	45.8	221.3M	45.92	119h - 97h 95h	97d
B2	4 723K × 4 754K	-2	231.9M	-0	48.8	228.2M	49.10	- 95h 93h 92h	95d
B1.1.1	4 661K × 4 670K	-2	231.2M	+7	49.5	229.3M	49.60	115h - 93h 91h	95d
B3	4 503K × 4 569K	-2	247.5M		54.2	239.0M	53.06	- 90h - -	-

Table 9. R211 matrices

Experiment series B achieved smaller matrices than A. The reason must be the different `keep` values during the pruning stage. Experiment series A kicked out 7.4M relations with the clique algorithm whereas B kept all the excess relations, performed more merges and discarded more relations during the merge steps. We can conclude that for this data the best thing was to skip the clique

algorithm. This is strongly connected to the fact that we barely had enough relations. Sieving any longer would surely have led to smaller matrices.

Matrix A1.1.1 performed better than matrix B1.1, which may seem counter-intuitive since B1.1 produced the smaller and lighter matrix. However, matrix A1.1.1 contained fewer rows (fewer prime ideals) than matrix B1.1 and due to the default truncation taking place in the Block Lanczos algorithm the effective A1.1.1 matrix was smaller in size and weight than the effective B1.1 matrix.

At B2 we also tried `mergelevel` 8 while having $m_{max} = 7$. `maxdiscard` was reached already at shrinkage pass 9 (with 15 possible passes) when the allowed weight increase was 5 original relations. The final matrix was larger than B1.1.1. We had chosen `maxrels` too low. It was 9, compared to 18 in B1.1.1. With `maxrels` 10 we achieved the desired reduction (B3).

RSA-155

The 155-digit number RSA-155 (512 bits!) was factored on August 22, 1999. A total of 130.8M relations were collected from 12 different sites. 6.1M relations were removed in individual `mergelevel` 0 runs. Another 39.2M duplicates where removed in a `mergelevel` 0 run on the whole amount of data. All the experiments below started with the remaining 85.5M relations and its 0.2M free relations. Therefore, in contrast to the previous examples, the figures in the discarded column do not contain any duplicates. See Table 4 for details.

Matrix B2 was used for the factorization. It took 225 hours on the Cray.

experiment	<code>mergelevel</code>	filter			discarded			not merged
		<code>filmin</code>	<code>keep</code>	<code>maxdiscard</code>	<code>sets</code>	<code>maxrels</code>	<code>maxpass</code>	
A	1	10M	<code>keep</code>	1.7M	19.1M	66 593K	385K	-
A1	5(1-3)	10M	<code>370K</code>	11.0 12	7.1M	370K	15K	67K
B	1	10M	<code>keep</code>	2.0M	20.2M	65 531K	684K	-
B1	8	10M	600K	9.0 15	6.9M	603K	81K 1611K/764K	
B2	8	7M	670K	9.0 15	6.7M	672K	13K 1576K/716K	
B3	8	7M	670K	10.0 15	7.1M	366K	317K 1432K/744K	
B4	16	7M	670K	9.0 15	6.6M	690K	-5K 4130K/694K	
B5	16	7M	670K	10.0 15	6.8M	482K	193K 3797K/562K	
B6	18	7M	670K	10.0 15	6.3M	672K	n.a.	n.a.
C	1	10M	<code>keep</code>	3.0M	23.6M	62 092K	1682K	-
C1	8	10M	1670K	8.0 15	6.8M	1675K	7K 1710K/698K	
D	1	10M	<code>keep</code>	5.0M	30.3K	55 402K	3677K	-
D1	8	10M	3670K	7.0 15	7.1M	3698K	-20K 2118K/780K	

Table 10. RSA-155 filter runs

The experiments indicate that retaining more data (`keep` $\geq 3.0M$) after the pruning stage did not help to reduce the size of the matrix.

Experiments B4 and D1 discarded too many relation-sets which is recognizable from the negative excess.

In B2 merging was stopped at shrinkage pass 11, while $m = 6$. Since there were still many unmerged ideals in B2, we tried to make the matrix smaller by increasing `maxrels` in B3 which allows also relation-sets with 10 relations, which were deleted in test B2. But even after this run many potential merge candidates remained unmerged, although `maxdiscard` was not reached. This indicates that the weight increase of the merges was considered too high and the merges were subsequently not executed. Next, we tried `mergelevel` 16, which is the maximum prime ideal frequency you can have a merge with for $m_{max} = 7$. Some reduction was achieved (B4 and B5). Finally, we took $m_{max} = 8$ together with `mergelevel` 18 and `maxrels` 10. `maxdiscard` was reached during shrinkage pass 14, when $m = m_{max}$.

exp.	matrix size	%	weight	%	col.w.	w_{eff}	col.w.	Cray
B2	$6\,699K \times 6\,711K$	-5	417.1M	62.2	415.5M	62.0	218h	
B6	$6\,342K \times 6\,354K$	+7	445.3M	70.1	443.4M	69.9	213h	

Table 11. RSA-155 matrices

Matrix B6 is 5% smaller than B2 but also 7% heavier. With $C = 14$ we can expect to save $1 - \frac{14 \cdot 6.342^2 + 6.342 \cdot 445.3}{14 \cdot 6.699^2 + 6.699 \cdot 417.1} \approx 1\%$ running time, which is too small a gain to accept the weight increase, whereas with $C = 37$ or $C = 49$ we may save 3% or 4%, respectively. The effective runs on the Cray ($C = 37$) indicate a saving of 2%.

5 Conclusions

We extended our previous `filter` program to allow higher-way merges and proved theoretically and practically that we can reduce Block Lanczos running time by performing higher-way merges. We determined limits for the weight of pivot columns.

During a merge, instead of merging by pivoting we calculate a minimum spanning tree in order to assure minimum weight increase.

A denser matrix allows for more weight increase during a merge than a lighter one: this means we can merge with denser pivot columns. Therefore we do the light merges before the heavier ones.

We determined the ratio between the two terms characterizing the running time of Block Lanczos for different implementations. To which extent we can profit from higher-way merges depends on this ratio. We saw values ranging from 14 to 49. With the help of this constants we can estimate the running time of a matrix, given the running time of another matrix.

Collecting more data than necessary is advisable. The clique algorithm enables us to get rid of excess data quickly and in a sensible way. It is a useful tool when having abundant excess.

References

1. Hendrik Boender. *Factoring Large Integers with the Quadratic Sieve*. PhD thesis, Rijksuniversiteit Leiden, 1997.
2. Joe P. Buhler, Hendrik W. Lenstra, Jr., and Carl Pomerance. Factoring integers with the number field sieve. In Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors, *The development of the number field sieve*, number 1554 in Lecture Notes in Mathematics, pages 50–94. Springer-Verlag, 1993.
3. Stefania Cavallar, Bruce Dodson, Arjen K. Lenstra, Paul Leyland, Walter Lioen, Peter L. Montgomery, Brian Murphy, Herman te Riele, and Paul Zimmermann. Factorization of RSA-140 using the number field sieve. In Kwok Yan Lam, Eiji Okamoto, and Chaoping Xing, editors, *Advances in Cryptology - Asiacrypt '99*, volume 1716 of *Lecture Notes in Computer Science*, pages 195–207. Springer-Verlag, 1999.
4. Stefania Cavallar, Bruce Dodson, Arjen K. Lenstra, Paul Leyland, Walter Lioen, Peter L. Montgomery, Herman te Riele, and Paul Zimmermann. 211-digit SNFS factorization. Available from <ftp://ftp.cwi.nl/pub/herman/NFSrecords/SNFS-211>, April 1999.
5. Stefania Cavallar, Bruce Dodson, Arjen K. Lenstra, Walter Lioen, Peter L. Montgomery, Brian Murphy, Herman te Riele, Karen Aardal, Jeff Gilchrist, Gérard Guillerm, Paul Leyland, Joël Marchand, François Morain, Alec Muffett, Chris Putnam, Craig Putnam, and Paul Zimmermann. Factorization of a 512-bit RSA modulus. Submitted to Eurocrypt 2000.
6. James Cowie, Bruce Dodson, R.-Marije Elkenbracht-Huizing, Arjen K. Lenstra, Peter L. Montgomery, and Jörg Zayer. A world wide number field sieve factoring record: on to 512 bits. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology - Asiacrypt '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 382–394. Springer-Verlag, 1996.
7. Thomas F. Denny. Solving large sparse systems of linear equations over finite prime fields. Transparencies of a lecture of the Cryptography Group at CWI, May 1995.
8. Reina-Marije Elkenbracht-Huizing. An implementation of the number field sieve. *Experimental Mathematics*, 5(3):231–253, 1996.
9. Ronald L. Graham and Pavol Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, January 1985.
10. Jürgen Neukirch. *Algebraische Zahlentheorie*. Springer-Verlag, 1992.
11. Donald E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial computing*. Addison-Wesley, 1993.
12. Donald E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, second edition, 1998.
13. Brian A. LaMacchia and Andrew M. Odlyzko. Solving large sparse linear systems over finite fields. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology - Crypto '90*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133. Springer-Verlag, 1991.
14. Serge Lang. *Algebraic Number Theory*. Springer, 1986.

15. Peter L. Montgomery. Square roots of products of algebraic numbers. In W. Gautschi, editor, *Mathematics of Computation 1943–1993: a Half-Century of Computational Mathematics*, volume 48 of *Proceedings of Symposia in Applied Mathematics*, pages 567–571. American Mathematical Society, 1994.
16. Peter L. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology - Eurocrypt '95*, volume 921 of *Lecture Notes in Computer Science*, pages 106–120. Springer-Verlag, 1995.
17. Phong Nguyen. A Montgomery-like square root for the number field sieve. In J. P. Buhler, editor, *Algorithmic Number Theory - ANTS-III*, volume 1423 of *Lecture Notes in Computer Science*, pages 151–168. Springer-Verlag, 1998.
18. J. M. Pollard. The lattice sieve. In Arjen K. Lenstra and Hendrik W. Lenstra, Jr., editors, *The development of the number field sieve*, number 1554 in Lecture Notes in Mathematics, pages 43–49. Springer-Verlag, 1993.
19. Carl Pomerance and J. W. Smith. Reduction of huge, sparse matrices over finite fields via created catastrophes. *Experimental Mathematics*, 1(2):89–94, 1992.

Factoring Polynomials over Finite Fields and Stable Colorings of Tournaments

Qi Cheng and Ming-Deh A. Huang

Computer Science Department
University of Southern California
`{qcheng,huang}@cs.usc.edu`

Abstract. In this paper we develop new algorithms for factoring polynomials over finite fields by exploring an interesting connection between the algebraic factoring problem and the combinatorial problem of stable coloring of tournaments. We present an algorithm which can be viewed as a recursive refinement scheme through which most cases of polynomials are completely factored in deterministic polynomial time within the first level of refinement, most of the remaining cases are factored completely before the end of the second level refinement, and so on. The algorithm has average polynomial time complexity and $(n^{\log n} \log p)^{O(1)}$ worst case complexity. Under a purely combinatorial conjecture concerning tournaments, the algorithm has worst case complexity $(n^{\log \log n} \log p)^{O(1)}$. Our approach is also useful in reducing the amount of randomness needed to factor a polynomial completely in expected polynomial time. We present a random polynomial time algorithm for factoring polynomials over finite fields which requires only $\log p$ random bits. All these results assume the Extended Riemann Hypothesis.

1 Introduction

It is well-known that polynomials over finite fields can be factored in random polynomial time [5,11]. However all attempts for finding a deterministic polynomial time algorithm for the problem have yielded only partial results so far. These results can be classified into two categories: those assuming the *Generalized Riemann Hypothesis* (GRH) or the *Extended Riemann Hypothesis* (ERH) and those that do not rely on any unproven assumptions. In general the unconditional results are more restrictive than those assuming the GRH or the ERH, yet they are more difficult to come by. Assuming the GRH, the best result so far is a deterministic algorithm due to Evdokimov [7] which factors polynomials of degree n over a finite field F_q in $(n^{\log n} \log q)^{O(1)}$ time. This result improves on the result of Ronyai[13] which solves the factoring problem in polynomial time when the degree of the input polynomial is bounded by a constant. We refer the readers to [4] for an extensive survey of other results and earlier works concerning deterministic factorization of polynomials over finite fields and related problems such as deterministic construction of finite fields and finite field isomorphism problems.

In this paper we develop new algorithms for factoring polynomials over finite fields by exploring an interesting connection between the algebraic factoring problem and the combinatorial problem of stable coloring of tournaments. In this approach we associate a polynomial to be factored with a tournament on its roots. We design algebraic procedures that explore symmetry in the associated tournament and cause the polynomial to split into factors according to the symmetry classes. Further splitting, if necessary, is effected as deeper levels of symmetry is explored through algebraic means. The resulting algorithm can be viewed as a recursive refinement scheme through which most cases of polynomials are split completely at the first level within polynomial time, most of the remaining cases are split completely before the end of the second level refinement, and so on.

The first level of our refinement scheme is a procedure which uses Ronyai's method [13] as building blocks. The basic observation is that Ronyai's method, when applied to a polynomial, groups the roots of the polynomial into factors according to scores in the associated tournament. By refining the method we obtain a procedure which implicitly performs stable coloring on the tournament. As combinatorial theory shows that most graphs decompose into singletons under a stable coloring, we can similarly show that most polynomials decompose into linear factors under this procedure. Should a non-linear factor survive the first level of refinement, it will be passed to higher levels of refinement where algebraic procedures are employed to explore higher levels of stable coloring on the tournament.

The resulting deterministic algorithm has average polynomial time complexity and $P(n^{\log n}, \log p)$ worst case complexity, on input polynomials of degree n over F_p . Moreover, all but at most $2^{-n/5}$ fraction of cases exit at the first level of the refinement scheme being completely factored. Then most of the remaining cases are split at the next level of refinement, and so on. The amount of time in going through the i -th level of refinement is bounded by $P(n^i, \log p)$ where $P(.,.)$ denotes a polynomial function. All cases are completely factored after no more than $\log n / 1.5$ levels of refinement.

The result assume the existence of quadratic nonresidues modulo p with absolute value polynomially bounded in $\log p$, which is true assuming the ERH. In bounding the fraction of cases that may need to go on to higher level of refinement we need the additional assumption that $n \leq \log p / 2$.

The tournament approach is also useful in reducing the amount of randomness needed to factor a polynomial completely in expected polynomial time. We show that there is a random polynomial time algorithm for factoring polynomials over finite fields which requires only $\log p$ random bits. This is a significant reduction from the $O(n \log p)$ random bits required by Berlekamp's method.

We will concentrate on polynomials all of whose roots are distinct and in a prime field F_p . The result can be extended to the general case by a well-known polynomial time reduction from the general case to the case we consider. We refer to [4] for details about this reduction. We assume the ERH throughout the rest of this paper.

Under a purely combinatorial conjecture concerning tournaments, we can show that the maximum level of refinement in our algorithm is as low as $\log \log n$, which implies that our algorithm has worst case complexity $(n^{\log \log n} \log p)^{O(1)}$. There are strong evidences for the conjecture which we discuss in Section 4.

We remark that the method in this paper can be used to prove that polynomials of degree n over F_p can be factored completely in time $P(n^{\delta(p)}, \log p)$, where P is a polynomial function and $\delta(p)$ is the size of largest transitive subgraph in the multicolor cyclotomic tournament over F_p . In light of this, an interesting open problem is to derive a sharp upper bound for $\delta(p)$.

2 The First Level of Refinement

Let $f \in F_p[x]$ be a polynomial with all roots distinct and in F_p . We construct tournaments on F_p as follows. First assume that $p \equiv 3 \pmod{4}$. For $a, b \in F_p$, a has an arc to b iff $a - b$ is a quadratic non-residue. This tournament is called *quadratic residue tournament* or *Paley tournament* [12]. Let

$$f = (x - a_1)(x - a_2) \cdots (x - a_n).$$

We associate with f the subtournament induced by a_1, \dots, a_n in Paley tournament. The *score* of a root a_i is the number of roots dominated by a_i . Define the *score polynomial* of f , denoted $S(f)$, as

$$\prod_{i=1}^n (x - a_i)^{b_i} \tag{1}$$

where b_i is the score of a_i .

When $p \equiv 1 \pmod{4}$, suppose $p - 1 = 2^k r'$, where r' is odd. We construct a *multicolor* tournament over F_p called the *cyclotomic tournament* over F_p such that for $i, j, s, t \in F_p$ with $i \neq j, s \neq t$, the arc (i, j) has the same color as the arc (s, t) if and only if $(i - j)r' = (s - t)r'$. We associate to f the sub-cyclotomic tournament on the roots of f . We define the score polynomial of f with respect to an arc color similar to above. (Actually we can replace 2^k by the smooth part of $p - 1$, and define generalized cyclotomic tournament.)

An interesting observation is that the score polynomial $S(f)$ can be computed in polynomial time using Ronyai's method. We outline below how this can be done. For ease of presentation we assume $p \equiv 3 \pmod{4}$.

Let A be the companion matrix of f . Following Ronyai we construct the following linear space V spanned by $\mu_i \otimes \mu_j$, $i \neq j$, where μ_1, \dots, μ_n are characteristic vectors of A , that is

$$A\mu_i = a_i\mu_i.$$

The linear transformation

$$H = I \otimes A - A \otimes I$$

acts upon V with $a_i - a_j$ as eigenvalues for $i \neq j$. The characteristic polynomial of $C = H^{r'}$ where $r' = (p - 1)/2$ has -1 as a root. The invariant subspace of $C + 1$ contains all basic tensors of the form $\mu_i \otimes \mu_j$, $i \neq j$ where $a_i - a_j$ is a quadratic nonresidue; that is, a_i dominates a_j . Hence as we construct this invariant subspace and construct the characteristic polynomial of the action of $A \otimes I$ on the subspace we get $S(f)$.

Theorem 1. *There is an algorithm that given a polynomial f in $F_p[x]$ of degree n , computes the score polynomial $S(f)$ in polynomial time.*

This is a special case of Theorem 4, which we will introduce and prove later.

From f and $S(f)$ we can split f into factors each having roots of the same score with the following procedure.

Algorithm 1. *Input f with distinct roots in F_p .*

1. Calculate $S(f)$ with respect to one of arc color (using Ronyai's algorithm).
2. Let $f_1 = S(f)$, $f_2 = f$.
3. While $f_2|f_1$ do $f_1 = f_1/f_2$.
4. If $f_1 = 1$, quit the algorithm, otherwise output $f_2/\gcd(f_1, f_2)$.
5. let $f_2 = \gcd(f_1, f_2)$. go to 3.

We call a tournament *regular* if every vertex dominates the same number of vertices. For an irregular polynomial (tournament), after we apply the algorithm, we get several factors corresponding to the scores. However, we need not stop here. Suppose a factor is not regular (i.e. the roots of the factor do not induce a regular subtournament). Then it will be split when the algorithm is applied to it. Applying the algorithm to the product of two factors may also cause further splitting. These ideas lead to the following refinement procedure on a set of factors with disjoint sets of roots.

Algorithm 2. *Input a set of relatively prime polynomials $\{f_1, f_2, \dots, f_n\}$, each with distinct roots in F_p .*

1. For $1 \leq i \leq n$, apply the algorithm (1) on f_i , let the set of output polynomials be S_i .
2. For $1 \leq i < j \leq n$, apply algorithm (1) on $f_i f_j$, for every output factor g , put $\gcd(g, f_i)$ into S_i , $\gcd(g, f_j)$ into S_j .
3. For every S_i , if there are any two polynomial $g, h \in V_i$, such that $\gcd(g, h) \neq 1$, then remove g, h from S_i and add $\gcd(g, h), g/\gcd(g, h), h/\gcd(g, h)$ into S_i .

We apply Algorithm (1) to f and then apply Algorithm (2) to the set of factors output by Algorithm (1). As we observe what is happening to the underlying tournament, we find that the process is very similar to the elementary refinement for undirected graphs[3]. The first procedure partitions the roots by score. Suppose C_1, C_2, \dots, C_h form the partition. For all roots x , let $N_i(x)$ denote the number of neighbors of x in C_i . In applying Algorithm (2) to the corresponding

set of factors, we first apply Algorithm (1) on C_i . This amounts to comparing $N_i(x)$ for all $x \in C_i$. Then we apply Algorithm (1) on $C_i C_j$. This amounts to comparing $N_i(x) + N_j(x)$ for all $x \in C_i \cup C_j$. When we exit Algorithm (2), we have refined the partition in the following manner. Two roots x, y are now in the same class iff they are in same class before the refinement, and

$$(N_1(x), N_2(x), \dots, N_h(x)) = (N_1(y), N_2(y), \dots, N_h(y)).$$

After we repeat Algorithm (2) at most n times, we will reach a point where the partition remains unchanged. At this point the partition of the roots is a *stable coloring* in the following sense.

Definition 1. A partition of the vertex set of a tournament be into vertex class C_1, \dots, C_m , is a **level-one stable coloring** if

1. C_i , $1 \leq i \leq m$, induces a regular subtournament,
2. For $1 \leq i, j \leq m$, for all $u, v \in C_i$, u dominates the same number of vertices in C_j as v does.

At this point we have completed the description of the first level of refinement in our algorithm for factoring polynomials over F_p . It is interesting to observe that after the level-one refinement, each factor is the union of some vertex orbits under the automorphism group of the tournament.

Babai and Kucera proved in [3] that almost all graphs can be decomposed to singletons by only two refinement steps. We can prove a similar result for tournaments.

Lemma 1. Let T be a random tournament on n vertices selected from the uniform distribution over the set of labeled n -tournaments. The probability that T cannot be factored into singletons by the refinement is less than $2^{-n/5}$.

See [6] for proof of the lemma. Based on the lemma and [8] we prove the following

Theorem 2. The fraction of polynomials over F_p with degree $n < \log p/2$ that cannot be split completely by the first level of refinement is less than $2^{-n/5}$.

Proof. If a separable polynomial f has all roots on F_p , its root set will induce a sub-tournament in Paley tournament. On the other hand, every induced sub-tournament in Paley tournament corresponds to a completely splitting separable polynomial over F_p . It was proved in [8] that every labeled tournament (graph) of order n occurs roughly as frequently as it should as induced subtournament (sub-graph) in Paley tournament (graph), namely, with probability $(1 + o(1))/2^{\binom{n}{2}}$, when $n < (\log p)/2$. Hence the theorem follows from Lemma 1.

It can be shown that if the first level refinement cannot split $f(x) = \prod(x - a_i)$ with degree $n < \sqrt{\log p}$ completely, then the probability that it cannot split $\prod(x - (a_i + k)^2)$ completely is less than 2^{-cn} , if k is uniformly picked up from F_p . From this we prove

Theorem 3. *There is a randomized algorithm using only $\log p$ many random bits to split a polynomial of degree n over F_p completely in expected polynomial time when $n < \sqrt{\log p}$.*

Proof. $(a_i + k)^2 - (a_j + k)^2 = (a_i - a_j)(a_i + a_j + 2k)$. Fix a_1, a_2, \dots, a_n , w.l.o.g, assume for any $i < j, u < v, a_i + a_j \neq a_u + a_v$ except when $i = u, j = v$. Let k be a random variable, uniformly taking value from set

$$\{k \mid \text{for any } i \neq j, a_i + a_j + 2k \neq 0\}.$$

Let $\chi : F_p^* \rightarrow \{1, -1\}$ be the character which sends x to $x^{\frac{p-1}{2}}$, then $\chi(a_1 + a_2 + 2k), \chi(a_1 + a_3 + 2k), \dots, \chi(a_{n-1} + a_n + 2k)$ is a random $1, -1$ sequence with size $\log p/2$ with uniformly distribution[8]. Hence $(a_1 + k)^2, (a_2 + k)^2, \dots, (a_n + k)^2$ induce all subtournaments of the Paley tournament with uniformly distribution.

3 The Second and Higher Levels of Refinement

A factor which remains after the first level of refinement has an underlying tournament which is regular. To refine it further we look for coherent stable colorings on all the subtournaments obtained by removing one root (vertex) from the factor (tournament).

Definition 2. *Let C_1, C_2, \dots, C_n be a level-one stable coloring for a tournament T , C'_1, C'_2, \dots, C'_m be a level-one stable coloring for a tournament T' , we say the two coloring are coherent if*

- $n = m$ and $|C_i| = |C'_i|$, for all $1 \leq i \leq n$.
- For any arc color E and $i \neq j$, if every vertex in C_i has k E -arcs to C_j , then every vertex in C'_i has k E -arcs to C'_j .

Definition 3. *Suppose T is a regular tournament with vertices v_1, \dots, v_n . Suppose $C_1^{v_i}, C_2^{v_i}, \dots, C_{m_i}^{v_i}$ is a level-one stable coloring of $T - v_i$. We say that the collection of these level-one stable colorings constitutes a level-two stable coloring for T , if they are coherent with one another, and for $1 \leq i \leq n$ and $1 \leq j \leq m_i$, either v_i dominates all vertices in $C_j^{v_i}$, or v_i is dominated by all the vertices in $C_j^{v_i}$.*

Below we describe how a regular polynomial can be manipulated algebraically so that either a level-two stable coloring on the underlying tournament is identified, or the polynomial is split.

In general let f be a polynomial of degree n with distinct roots a_1, \dots, a_n in F_p as before. Let $R = F_p[x]/(f) = F_p[A]$, where $A = x \bmod f$. Let $f^* \in R[x]$ so that

$$f(x) = (x - A)f^*.$$

There exist uniquely determined primitive idempotents $e_i \in R$, $1 \leq i \leq n$, such that $\sum_{i=1}^n e_i = 1$, $e_i e_j = e_i \delta_{ij}$. In fact, $e_i = \prod_{j \neq i} (A - a_j) / \prod_{j \neq i} (a_i - a_j)$.

For every element $c \in R$, there exist unique elements c_1, \dots, c_n such that $c = \sum_{i=1}^n c_i e_i$. We call c_i the i th canonical projection of c on F_p . The canonical projections of a polynomial in $R[x]$ can be similarly defined. In particular,

$$f^* = \sum_{i=1}^n f_i e_i \text{ where } f_i = \prod_{j \neq i} (x - a_j).$$

We remark that since f_i represents the subtournament obtained from the tournament of f by removing the root a_i , f^* succinctly represents all these subtournaments simultaneously.

An element of R has the form $h(A)$ where h is a polynomial over F_p of degree less than n . It is a zero-divisor in R iff the GCD of h and f is not 1. In other word as we attempt to find an inverse of $h(A)$ in R by computing the GCD of $h(x)$ and $f(x)$, we either succeed or find a nontrivial factor of f .

We can extend Ronyai's algorithm to work on a polynomial over general completely splitting algebra over finite field. For definition of *completely splitting semisimple algebras over finite fields* and *completely splitting polynomials* over such algebras, see [7]. Let R be completely splitting semisimple algebra of dimension m over F_p . Denote the uniquely determined primitive idempotents as $e_i \in R$, $1 \leq i \leq m$. By definition we have $\sum_{i=1}^n e_i = 1$, $e_i e_j = e_i \delta_{ij}$, and

$$R \cong \bigoplus_{1 \leq i \leq m} F_p.$$

We may not know these idempotents at the begin of the algorithm.

For any $g(x) \in R[x]$, if $g(x) = \sum_{i=1}^m g_i(x) e_i$, where $g_i(x) \in F_p[x]$, $1 \leq i \leq m$ and e_1, \dots, e_m are the primitive idempotents of R over F_p . Define the *score polynomial* of $g(x)$ by

$$S(g(x)) = \sum_{i=1}^m S(g_i(x)) e_i.$$

Thus $S(g)$ succinctly represents the set of score polynomials $S(g_i)$ for $i = 1, \dots, m$.

In the following theorem we assume: (1) The ring operations can be carried out in polynomial time. (By polynomial time, we mean in time $(m \log p)^{O(1)}$). (2) Given $a \in R$, we can determine whether a is a zero divisor, and if not, find its inverse in polynomial time. (3) Given an l -th non-residue in the field F_p ($l|p-1$), if $a^{(p-1)/l}$ is an idempotent of algebra R , at least l distinct l -th roots of a can be found in time $(ml \log p)^{O(1)}$.

Theorem 4. Suppose R is ring with above properties. Let $f \in R[x]$ be completely splitting separable monic polynomial with degree n . There is a deterministic algorithm which in $(nm \log p)^{O(1)}$ time either finds a nontrivial zero-divisor in R (hence a nontrivial factor of f), or computes $S(f)$.

Proof. Let $f = \sum_{i=0}^n a_i x^i \in R[x]$, $a_n = 1$ be a monic polynomial, suppose

$$f = \sum_{1 \leq i \leq m} f_i(x) e_i,$$

where for any $1 \leq i \leq n$, $f_i(x) \in F_p[x]$ splits completely over F_p into n distinct linear factors.

Let A be the companion matrix of f ,

$$A = \sum_{1 \leq i \leq m} A_i e_i,$$

where A_i 's are $n \times n$ matrices over F_p . Let $a_1^{(i)}, a_2^{(i)}, \dots, a_n^{(i)}$ be the eigenvalues of A_i and $\mu_1^{(i)}, \mu_2^{(i)}, \dots, \mu_n^{(i)}$ be the corresponding characteristic vectors.

Consider the following linear transformation

$$G = I \otimes A - A \otimes I = \sum_{1 \leq i \leq m} e_i(I \otimes A_i) - e_i(A_i \otimes I),$$

which acts on R^{n^2} . The vectors in $L = G(R^{n^2})$ must have the form

$$\sum_{j,k,j \neq k} \sum_i c_{i,j,k} \mu_j^{(i)} \otimes \mu_k^{(i)} e_i = \sum_{j,k,j \neq k} (\sum_i c_{j,k,i} e_i) (\sum_i \mu_j^{(i)} \otimes \mu_k^{(i)} e_i).$$

It is a free module, having a basis $\{\sum_{1 \leq i \leq m} \mu_j^{(i)} \otimes \mu_k^{(i)} e_i | j \neq k, 1 \leq j, k \leq n\}$. The dimension of L is $n(n-1)$.

Let H be the transformation of G on L . Let the characteristic polynomial of $C = H^{r'}$ be $c(x) = \sum_i c_i(x) e_i$. Let α be one of roots of $c(x)$. In case $p \equiv 3 \pmod{4}$, $c(x)$ should be $(x-1)^{\frac{n(n-1)}{2}}(x+1)^{\frac{n(n-1)}{2}}$, that is to say, $\alpha \in \{1, -1\}$.

Let T be the kernel of $C - \alpha I$ as a linear map on L . The vector in T must have form

$$\sum_i \sum_{j,k, a_j^{(i)} - a_k^{(i)} \in \beta(F_p^*)^r} c_{i,j,k} \mu_j^{(i)} \otimes \mu_k^{(i)} e_i,$$

where β is one of r' -th roots of α (If $p \equiv 3 \pmod{4}$, $\alpha = -1$, β is any quadratic non-residue). T is a free module over R of dimension $\frac{n(n-1)}{2}$.

Let U be the transformation $A \otimes I$ on T . The characteristic polynomial of U is score polynomial (with respect to α).

Now we describe the algorithm to compute the score polynomial. The companion matrix of f is

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{pmatrix}$$

We compute $I \otimes A$ and $A \otimes I$, using Kronecker product of matrices. Then we construct linear space $L = G(R^{n^2})$ by doing Gauss elimination on $G = I \otimes A - A \otimes I$. In the process, we either encounter a zero divisor on R , or end

up with a basis for this linear space. Notice that R is a commutative ring with identity, hence it has invariant dimension property. This implies that the number of independent generators we get should be $n(n - 1)$. Notice that $C = H^{r'}$ can be computed by the squaring technique. If $p \equiv 1 \pmod{4}$, we need to factor $c(x)$ using Evdokimov's algorithm. We then need to construct a basis for T , the kernel of $C - \alpha I$, and compute U , the matrix for the transformation $A \otimes I$ on T . These tasks can be done using standard linear algebra methods. In the last step, we calculate the characteristic polynomial of U . We either get a zero divisor in the process, or obtain the score polynomial. The whole process takes polynomial time. The theorem follows.

With the above theorem we are in a position to extend Algorithms (1) and (2) in a natural way to completely splitting polynomials in $R[x]$, where R is a completely splitting algebra over F_p satisfying the conditions in Theorem 4. The steps in the algorithms which involve polynomial division or GCD need some modification due to the presence of zero divisors in R . To divide a polynomial g by a polynomial h over R , we need to check whether the leading coefficient of h is a zero divisor, and if not, to find its inverse in R . Encountering a zero divisor causes an early exit in the computation.

Following [9], the GCD of two polynomials over R can be defined as follow. Let $f(x), g(x) \in R[x]$ with $f(x) = \sum_{i=1}^n f_i(x)e_i, g(x) = \sum_{i=1}^n g_i(x)e_i$, where $f_i(x), g_i(x) \in F_p[x], 1 \leq i \leq n$. Define $\text{GCD}(f, g) = \sum_{i=1}^n \text{GCD}(f_i, g_i)e_i$. For $f, g \in R[x]$, $\text{GCD}(f, g)$ can be computed in polynomial time [9].

We call a polynomial $g \in R[x]$ *regular* if $g(x) = \sum_{i=1}^n g_i(x)e_i$ where $g_i \in F_p[x]$ is regular for all i . The above discussion shows that we can extend the level-one refinement in a natural way to a regular polynomial g over a ring R satisfying the conditions in Theorem 4. We call this the level-two refinement on g .

Theorem 5. *There is a deterministic polynomial time algorithm which on input a regular $f \in F_p[x]$ of degree n , either finds a nontrivial factor of f , or succinctly finds a level-two stable coloring for the tournament of f in the sense that it factors f^* as*

$$f^* = \prod_{i=1}^m g_i \text{ where } g_i = \sum_{j=1}^n C_i^{v_j} e_j$$

where $C_1^{v_j}, \dots, C_m^{v_j}$ constitute a level-one stable coloring \mathcal{C}_j for the subtournament obtained by removing the j -th root from f and $\mathcal{C}_1, \dots, \mathcal{C}_n$ form a level-two stable coloring for the tournament of f . Furthermore, $m \geq 2$, and (after reordering if necessary) there is $1 \leq r < m$, $\sum_{1 \leq i \leq r} \deg(g_i) = \sum_{r+1 \leq i \leq m} \deg(g_i) = \frac{n-1}{2}$.

Proof. Let T be the regular tournament associated with f . Let $O_T(x) = \{v \in T | x \text{ dominate } v\}$ and $I_T(x) = \{v \in T | v \text{ dominates } x\}$. We will omit the subscripts when there is no risk of confusion. For a subset S of $V(T)$, we denote the polynomial $\prod_{s \in S} (x - s)$ simply by S . Thus

$$f^*(x) = \sum_{i=1}^n (T - a_i)e_i.$$

If the tournament with n vertices is regular, then for any vertex x , every vertex in $O(x)$ has score $(n-1)/2$ in $T-x$, while every vertex in $I(x)$ has score $(n-3)/2$. So

$$S(f^*) = \sum_{i=1}^n O(a_i)^{(n-1)/2} I(a_i)^{(n-3)/2} e_i. \quad (2)$$

Let's examine what happens as we apply the level-two refinement on f^* . First we apply Algorithm (1) on input f^* . In case no early exit occurs, then from Eq. (2) we see that f^* is factored as $f_O^* f_I^*$ where

$$f_O^* = \sum_{i=1}^n O(a_i) e_i \text{ and } f_I^* = \sum_{i=1}^n I(a_i) e_I.$$

Then as we apply Algorithm (2) on input $\{f_O^*, f_I^*\}$, we may encounter a zero-divisor in $F_p[A]$ and exit the refinement early with f split as result. If we successfully run through the refinement without an early exit, then a level-one stable coloring has been found for each T_i simultaneously. Moreover, not encountering a zero divisor means that these level-one stable coloring are coherent(notice that during the computation if we obtain a polynomial $g = \sum_i g_i(x) e_i$, such that two of the component polynomials have different degrees, then the coefficient of the highest order term of g is a zero divisor.), thus a level-two stable coloring has been succinctly constructed in the factors of f^* over R . The rest of the assertions follows from the fact that f_O^* is of degree $\frac{n-1}{2}$. Finally the polynomial time bound follows from Theorem 4.

A tournament is called *doubly-regular* if it is regular and for every vertex v , the subtournaments induced on $O(v)$ and $I(v)$ are regular. For a doubly regular polynomial f , f^* may go through level-two refinement being factored only into f_O^* and f_I^* .

Define the *score vector* of a tournament as the sorted list of all the scores in the tournament. We call a tournament pseudo-vertex-symmetric if the score vector is the same for the subtournaments induced on $O(v)$ (respectively $I(v)$) for every vertex v [1]. Intuitively speaking, pseudo-vertex-symmetric tournaments are rare.

From the proof of Theorem 5, we can also conclude

Corollary 1. *If f has level-two stable coloring, it corresponds to a pseudo-vertex-symmetric tournament. If f^* has only two factors, f corresponds to doubly-regular tournament in Paley graph.*

The proof of Theorem 5 can be extended in a natural way to show that there is a deterministic polynomial time algorithm which on input a regular $f \in R[x]$ of degree n , where R is a ring satisfying the conditions in Theorem 4, either finds a nontrivial zero divisor of R , or succinctly and simultaneously finds a level-two stable coloring for each canonical projection of f .

Suppose f is regular with degree n . Let $R_0 = F_p$, $R_1 = R_0[x]/f(x)$. Applying Theorem 5 to f , we either find a nontrivial factor of f or split f^* over R_1 .

Suppose f^* is split over R_1 . Let f_1 be the factor with least degree n_1 . If $n_1 = 1$, then we can construct a zero-divisor in R_1 [7], hence factor f . Otherwise f_1 is a polynomial whose canonical projections are regular polynomials with order n_1 . Let $R_2 = R_1[x]/f_1(x)$. Then R_2 is a completely splitting algebra over F_p satisfying the conditions in Theorem 4 [7].

Let $f_1 = \sum_{i=1}^n T_i e_i$ where $T_i = \prod_{1 \leq j \leq n_1} (x - a_j^{(i)})$. Note that we then have $f_1 = \prod_{1 \leq j \leq n_1} (x - \sum_{1 \leq i \leq n} a_j^{(i)} e_i)$. Let $A = x \bmod f_1(x)$ and $f_1 = (x - A)f_1^*$. The idempotents over R_1 are

$$\begin{aligned} e_j^* &= \prod_{k, k \neq j} (A - \sum_{1 \leq i \leq n} a_k^{(i)} e_i) / \prod_{k, k \neq j} (\sum_{1 \leq i \leq n} a_j^{(i)} e_i - \sum_{1 \leq i \leq n} a_k^{(i)} e_i) \\ &= \sum_{1 \leq i \leq n} (\prod_{k, k \neq j} (A - a_k^{(i)})) / \prod_{k, k \neq j} (a_j^{(i)} - a_k^{(i)}) e_i \end{aligned}$$

Let $\epsilon_j^{(i)} = \prod_{k, k \neq j} (A - a_k^{(i)}) / \prod_{k, k \neq j} (a_j^{(i)} - a_k^{(i)})$. Then R_2 's canonical primitive idempotents are $\{\epsilon_j^{(i)} e_i | 1 \leq i \leq n, 1 \leq j \leq n_1\}$, and

$$f_1^* = \sum_{i=1}^n \sum_{j=1}^{n_1} (T_i - a_j^{(i)}) \epsilon_j^{(i)} e_i.$$

We apply Algorithm (1) and (2) on f_1^* over R_2 . Either an early exit leads to the splitting of f_1 over R_1 or even the splitting f over F_p ; or a level-two stable coloring is simultaneously found on every canonical projection of f_1 . Inductively, let f_i be the polynomial resulting from the i -th level of refinement with degree n_i over a completely splitting algebra R_i .

A tournament is called *triply-regular* if it is regular and for every vertex v , the subtournaments induced on $O(v)$ and $I(v)$ are doubly-regular. It is a remarkable fact that there is no triply-regular tournament with $n \geq 4$ vertices[10].

Theorem 6. *For any polynomial $f \in F_p[x]$ of degree n , the number of levels that our algorithm go through in order to factor f completely is at most $\frac{\log n}{1.5}$.*

Proof. We know that $n_{i+1} \leq n_i/2$. If the canonical projections of f_i are not doubly-regular, then $n_{i+1} \leq n_i/4$, and $n_{i+2} \leq n_i/8$. Otherwise, if the projections of f_i are doubly-regular, it is possible that $n_{i+1} = n_i/2$, but then the projections of f_{i+1} will not be doubly-regular, since there is no nontrivial triply-regular tournament, hence $n_{i+2} \leq n_{i+1}/4$, thus we have $n_{i+2} \leq n_i/8$. Therefore $n_t \leq 1$ if $t > \frac{\log n}{1.5}$.

4 Discussion

In general suppose T is a tournament that admits a level-two stable coloring. Put two arcs uv and xy in the same class iff in the stable coloring of $T - u$ and $T - x$,

v and y are in corresponding classes (that is $v \in C_j^u$ and $y \in C_j^x$ for some j). For any arc class G , we call graph $B_G = (V, G)$ a *base graph* for T with respect to the level-two stable coloring of T . Suppose T is the underlying tournament for a regular polynomial f and the level-two stable coloring is represented by the factoring of f^* into the product of g_i^* as in the theorem above. Then the base graphs are in one-one correspondence with the factors g_i^* . Each base graph is a regular digraph and the set of arcs in a base graph is the union of some arc orbits in the tournament under the automorphism group of the tournament.

The bound on the number of levels in the above theorem seems to be far from being tight. Define a function β from set of tournaments to the set of natural numbers as follows. If a tournament T is not regular or doesn't have second level stable coloring, $\beta(T) = 0$. If T has second level stable coloring, $\beta(T)$ is the maximum of $\beta(\mathcal{C}, T)$ among all the level-two stable colorings \mathcal{C} of T , where $\beta(\mathcal{C}, T)$ is the number of arcs in a minimum base graph with respect to \mathcal{C} .

Conjecture 1. Suppose T is a regular tournament on n vertices that admits a level-two stable coloring. Then for any level-two stable coloring of T , $C_1^{v_1}, \dots, C_m^{v_n}$, there is a C_j^i , such that $\beta(C_j^i) = O(n^c)$, where $c < 2$ is a constant independent of T .

Intuitively, the coherence requirement should already make it difficult for all the C_j^i to have large minimum base graphs, if they could all have level-two stable colorings at all. The conjecture implies that our deterministic algorithm factor a polynomial of degree n over F_p completely within time $(n^{\log \log n} \log p)^{O(1)}$. The fact that there is no triply-regular tournament with more than three vertices and the following observation of Babai [2] provide strong evidences for the conjecture.

Proposition 1. Let T be a vertex-transitive tournament with $n > 1$ vertices. Let v_0 be a vertex of T . Then for every vertex $v_1 \neq v_0$ there exists a vertex $v_2 \neq v_0, v_1$ such that the size of the orbit of the pair (v_1, v_2) in the stabilizer of v_0 is at most $(n - 1)/2$.

Another way to improve our results is to look at the case when we have a lot of arc colors.

Definition 4. We call a tournament with n vertices transitive if there is a linear order of its vertices, v_1, v_2, \dots, v_n , such that for any i and color C , if v_i C -dominates v_{i+1} , then v_i C -dominates v_j for any $j > i$.

Denote $\delta(p)$ be the size of largest transitive subgraph in a cyclotomic tournament. Heuristically when the number of colors gets bigger, $\delta(p)$ should become smaller, even down to a constant. One can for example proves that a random tournament with n vertices and n^c ($c < 1$) colors has only constant size transitive subtournament. We can prove that the polynomial in F_p can be factored completely in time $P(n^{\delta(p)}, \log p)$, where P is a polynomial function.

Acknowledgment

We would like to thank Laci Babai for showing us Proposition 1 and for very interesting discussions.

References

1. Annie Astie-Vidal, Vincent Dugat, Autonomous parts and decomposition of regular tournaments, *Discrete Mathematics* 111, (1993), 27-36.
2. L Babai, *personal communication*, 1998
3. L. Babai, Ludik Kucera, Canonical labelling of graphs in linear average time, *FOCS'79*, 39-46
4. E Bach, Jeffrey Shallit, *Algorithmic Number theory, Vol I*, The MIT Press, 1996
5. E. R. Berlekamp, Factoring polynomials over large finite fields, *Math. Comp.* 24, (1970), 713-735.
6. Qi Cheng, Fang Fang, Kolmogorov Random Graphs Only Have Trivial Stable Colorings, submitted, 2000.
7. Sergei Evdokimov, Factorization of Polynomials over Finite Field in subexponential Time under ERH, *Lecture Notes in Computer Science* 877, 1994, 209-219
8. P. Frankl, V. Rodl, R.M. Wilson, The number of submatrices of a given type in a Hadamard matrix and related results, *J. of Combinatorial Theory, Series B*, 44, 1988
9. Shuhong Gao, Factoring polynomials over large prime fields, to appear in *J. of Symbolic Computation*.
10. Vladimir Muller, Jan Pelant, On Strongly Homogeneous Tournaments, *Czechoslovak Mathematical Journal*, 24(99) , 1974, 379-391
11. M. O. Rabin, Probabilistic algorithms in finite fields, *Siam J. Computing*, 9 (1980), 128-138.
12. K.B. Reid and L. W. Beineke, Tournaments, *Selected Topics in Graph Theory*, L. W. Beineke and Robin J. Wilson Eds. Academic Press., 1978
13. Lajos Ronyai, Factoring Polynomials over Finite Fields, *J. Algorithms*, 9 (1988), 391-400. An earlier version appeared in *STOC'87*, 132-137.

Computing Special Values of Partial Zeta Functions

Gautam Chinta¹, Paul E. Gunnells¹, and Robert Sczech²

¹ Dept. of Mathematics, Columbia University
New York, NY 10027, USA

² Dept. of Mathematics and Computer Science, Rutgers University
Newark, NJ 07102–1811, USA

Abstract. We discuss computation of the special values of partial zeta functions associated to totally real number fields. The main tool is the *Eisenstein cocycle* Ψ , a group cocycle for $GL_n(\mathbb{Z})$; the special values are computed as periods of Ψ , and are expressed in terms of generalized Dedekind sums. We conclude with some numerical examples for cubic and quartic fields of small discriminant.

1 Introduction

Let K/\mathbb{Q} be a totally real number field of degree n with ring of integers \mathcal{O}_K , and let $U \subset \mathcal{O}_K^\times$ be the subgroup of totally positive units. Let $\mathfrak{f}, \mathfrak{b} \subset \mathcal{O}_K$ be relatively prime ideals. Then the *partial zeta function* associated to these data is defined by

$$\zeta_{\mathfrak{f}}(\mathfrak{b}, s) := \sum_{\mathfrak{a} \sim \mathfrak{b}} N(\mathfrak{a})^{-s},$$

where $\mathfrak{a} \sim \mathfrak{b}$ means $\mathfrak{a}\mathfrak{b}^{-1} = (\alpha)$, where α is a totally positive number in $1 + \mathfrak{f}\mathfrak{b}^{-1}$. According to a classical result of Klingen and Siegel [10], the special values $\zeta_{\mathfrak{f}}(\mathfrak{b}, k)$ are rational for nonpositive integers k . Moreover, the values $\zeta_{\mathfrak{f}}(\mathfrak{b}, 0)$ are especially important because of their connection with the Brumer-Stark conjecture and the Leopoldt conjecture [7,6,3,8,11].

In [9], one of us (RS) gave a cohomological interpretation of these special values by showing that they can be computed in finite terms as periods of the *Eisenstein cocycle*. This is a cocycle $\Psi \in H^{n-1}(GL_n(\mathbb{Z}); \mathcal{M})$, where \mathcal{M} is a certain $GL_n(\mathbb{Z})$ -module. Then two of us (PEG and RS) showed in [5] that the Eisenstein cocycle is an effectively computable object. More precisely, using the cocycle one can express $\zeta_{\mathfrak{f}}(\mathfrak{b}, k)$ as a finite sum of *generalized Dedekind sums*, and the latter can be effectively computed by a continued-fraction algorithm that uses a generalization of the classical Dedekind-Rademacher reciprocity law.

In this note we describe an ongoing project to build a database of $\zeta_{\mathfrak{f}}(\mathfrak{b}, 0)$ for various fields K and ideals $\mathfrak{f}, \mathfrak{b}$. We recall the definition of the Eisenstein cocycle and its relation to the special values (§2), and discuss the effective computation of Dedekind sums (§3). We conclude with examples of special values for some fields of degree 3 and 4 (§4).

2 Dedekind Sums and the Eisenstein Cocycle

2.1

Let σ be a square matrix with integral columns $\sigma_j \in \mathbb{Z}^n$ ($j = 1, \dots, n$), and let $L \subset \mathbb{Z}^n$ be a lattice of rank $r \geq 1$. Let $v \in \mathbb{Q}^n$, and let $e \in \mathbb{Z}^n$ with $e_j \geq 1$. Then the *Dedekind sum* S associated to the data (L, σ, e, v) is defined by

$$S = S(L, \sigma, e, v) := \sum'_{x \in L} \mathbf{e}(\langle x, v \rangle) \frac{\det \sigma}{\langle x, \sigma_1 \rangle^{e_1} \cdots \langle x, \sigma_n \rangle^{e_n}}. \quad (1)$$

Here $\langle x, y \rangle := \sum x_i y_i$ is the usual scalar product on \mathbb{R}^n , $\mathbf{e}(t)$ is the character $\exp(2\pi i t)$, and the prime next to the summation means to omit terms for which the denominator vanishes. The series (1) converges absolutely if all $e_j > 1$, but may only converge conditionally if $e_j = 1$ for some j . In this latter case we can define the sum by the *Q -limit*

$$\sum'_{x \in L} a(x) \Big|_Q := \lim_{t \rightarrow \infty} \left(\sum'_{|Q(x)| < t} a(x) \right), \quad (2)$$

where Q is any finite product of real-valued linear forms on \mathbb{R}^n that doesn't vanish on $\mathbb{Q}^n \setminus \{0\}$. One can precisely determine how the value of (1) depends on Q ([9, Thm. 7]). The sum S is always a rational number times a power of $2\pi i$.

2.2

We recall now the definition of the Eisenstein cocycle Ψ and its relationship with the special values $\zeta_f(b, k)$. For simplicity, we describe only material necessary to compute the special value at $k = 0$, and refer to [9,5] for other k .

Let $\mathcal{A} = (A_1, \dots, A_n) \in (GL_n(\mathbb{R}))^n$ be an n -tuple of matrices. For an n -tuple $d = (d_1, \dots, d_n)$ of integers $1 \leq d_i \leq n$, let $\mathcal{A}(d) \subseteq \mathbb{R}^n$ be the subspace generated by all columns A_{ij} such that $j < d_i$. (Here A_{ij} denotes the j th column of the matrix A_i .) Writing $\mathcal{A}(d)^\perp$ for the orthogonal complement of $\mathcal{A}(d)$ in \mathbb{R}^n , we let

$$X(d) = \mathcal{A}(d)^\perp \setminus \bigcup_{i=1}^n \sigma_i^\perp, \quad \text{where } \sigma_i = A_{id_i}. \quad (3)$$

The n -tuple \mathcal{A} determines a decomposition of $\mathbb{R}^n \setminus \{0\}$ into linear strata

$$\bigsqcup_{d \in D} X(d), \quad (4)$$

indexed by the finite set

$$D = D(\mathcal{A}) = \{d \mid X(d) \neq \emptyset\}.$$

Associated to this decomposition is a collection of rational functions $\psi(\mathcal{A})$ on $\mathbb{R}^n \setminus \{0\}$, defined by

$$\psi(\mathcal{A})(x) = \frac{\det(\sigma_1, \dots, \sigma_n)}{\langle x, \sigma_1 \rangle \cdots \langle x, \sigma_n \rangle}, \quad \text{if } x \in X(d).$$

Note that $\psi(\mathcal{A})(x)$ is well-defined by the construction of $X(d)$.

Let $v \in \mathbb{R}^n$, and let Q be defined as in §2.1. Then the *Eisenstein cocycle* Ψ is defined as

$$\Psi = \Psi(\mathcal{A})(Q, v) := (2\pi i)^{-n} \sum_{x \in \mathbb{Z}^n} \mathbf{e}(\langle x, v \rangle) \psi(\mathcal{A})(x) \Big|_Q.$$

One can show that Ψ is a homogeneous $(n-1)$ -cocycle for $GL_n(\mathbb{Z})$. Furthermore, we can express Ψ in terms of Dedekind sums

$$\Psi(\mathcal{A})(Q, v) = (2\pi i)^{-n} \sum_{d \in D} S(L(d), \sigma, \mathbf{1}, v) \Big|_Q, \quad (5)$$

where σ is the matrix with columns A_{id_i} , ($i = 1, \dots, n$), $L(d)$ is the lattice $\mathcal{A}(d)^\perp \cap \mathbb{Z}^n$, and $\mathbf{1}$ is the vector $(1, \dots, 1)$.

2.3

Now we describe how Ψ can be used to compute special values. Let W be a \mathbb{Z} -basis for the fractional ideal $\mathfrak{f}\mathfrak{b}^{-1} = \sum \mathbb{Z}W_j$, and let W^* be the dual basis with respect to the trace form. Via the n real embeddings τ_i , $i = 1, \dots, n$, any $x \in K$ determines a row vector $(\tau_1(x), \dots, \tau_n(x))$. Hence we may identify W with a matrix in $GL_n(\mathbb{R})$: the j th row of this matrix is the image of the j th basis element of W . Let

$$Q(X) = \prod_i \sum_j X_j(\tau_i(W_j^*)),$$

and let $v \in \mathbb{Q}^n$ be defined by $v_j = \text{Tr}(W_j^*)$.

Let $\nu = n - 1$, and let $\varepsilon_1, \dots, \varepsilon_\nu$ be a basis for the totally positive units U . Using the regular representation ρ with respect to the basis W , we identify the units ε_j with elements $A_j = \rho(\varepsilon_j)^t \in GL_n(\mathbb{Z})$. Using the bar notation

$$[A_1 | \cdots | A_\nu] := (1, A_1, A_1 A_2, \dots, A_1 \cdots A_\nu) \in (GL_n(\mathbb{Z}))^n,$$

we have the following proposition expressing the zeta values in terms of the Eisenstein cocycle:

Proposition 1. [9,5] *Let $U_{\mathfrak{f}}$ be the subgroup $U \cap (1 + \mathfrak{f})$, and let π run through all permutations of $\{1, \dots, \nu\}$. Then*

$$\zeta_{\mathfrak{f}}(\mathfrak{b}, 0) = \eta \sum_{\varepsilon \in U/U_{\mathfrak{f}}} \sum_{\pi} \text{sgn}(\pi) \Psi([A_{\pi(1)} | \cdots | A_{\pi(\nu)}])(Q, \rho(\varepsilon)^t v).$$

Here $\eta = \pm 1$ is defined by

$$\eta = (-1)^\nu \text{sgn}(\det W) \text{sgn}(R),$$

where $R = \det(\log \tau_j(\varepsilon_i))$, $1 \leq i, j \leq \nu$.

3 Diagonality and Unimodularity

3.1

We define the *rank* of $S = S(L, \sigma, e, v)$ to be the rank of the lattice L . It is easy to see that after a $GL_n(\mathbb{Q})$ transformation, we may assume that L is the sublattice $Z^\ell \subset \mathbb{Z}^n$ spanned by the first ℓ standard basis vectors, where ℓ is the rank of L . Furthermore, by multiplying by an appropriate rational factor, permuting columns and repeating columns if necessary, we may assume the pair (Z^ℓ, σ) satisfies the following conditions:

- (i) For each column σ_j , the vector of the first ℓ components of σ_j is primitive and integral.
- (ii) If two columns of σ induce proportional linear forms on Z^ℓ , then these two linear forms coincide on Z^ℓ , and are adjacent columns of σ .
- (iii) The vector $e = \mathbf{1}$.

Let $S(Z^\ell, \sigma, \mathbf{1}, v)$ be a Dedekind sum satisfying the three conditions above. Let $\pi: \mathbb{R}^N \rightarrow \mathbb{R}^\ell$ be the projection on the first ℓ components, and let $\pi(\sigma)$ be the $\ell \times n$ matrix with columns $\pi(\sigma_i)$.

Definition 1. Let $M(\sigma)$ be the set of maximal minors of $\pi(\sigma)$. Then the index of S , denoted $\|S\|$, is defined to be

$$\max_{\tau \in M(\sigma)} |\det \tau|.$$

A Dedekind sum is unimodular if $\|S\| = 1$.

3.2

Now define a partition

$$[\![n]\!] = \bigsqcup_{k=1}^s I_k, \quad \ell \leq s \leq n \tag{6}$$

as follows. Put

$$i, j \in I_k \quad \text{if and only if} \quad \pi(\sigma_i) = \pi(\sigma_j).$$

In other words, two elements of $[\![n]\!]$ are in the same set of the partition if the corresponding columns of σ induce the same linear form on Z^ℓ .

Let $p_k = \#I_k$.

Definition 2. The vector $p(S) = (p_1, \dots, p_s)$ is called the type of S . A Dedekind sum is called diagonal if $p(S)$ has length ℓ .

3.3

The virtue of diagonality is that a diagonal Dedekind sum S may be evaluated as a finite sum of products of generalized Bernoulli polynomials. Furthermore, the number of terms in this finite sum is the index of S . Hence diagonal and unimodular Dedekind sums can be evaluated very rapidly.

In general, the Dedekind sums in (5) aren't diagonal. However, we have the following theorem, which is the main result of [5]:

Theorem 1. [5] *Every Dedekind sum $S(L, \sigma, e, v)$ can be expressed as a finite rational linear combination of unimodular diagonal sums. If n , $\text{Rank } L$, and e are fixed, then this expression can be computed in time polynomial in $\log \|S\|$. Moreover, the number of terms in this expression is bounded by a polynomial in $\log \|S\|$.*

The key ingredient in the proof of Theorem 1 is a “reciprocity law” for higher-dimensional Dedekind sums. For any nonzero point $v \in \mathbb{R}^n$, let v^\perp be the hyperplane $\{x \mid \langle v, x \rangle = 0\}$. Let Q be a finite product of real-valued linear forms on \mathbb{R}^n that do not vanish on $\mathbb{Q}^n \setminus \{0\}$.

Proposition 2. *Let $\sigma_0, \dots, \sigma_n \in \mathbb{Z}^n$ be nonzero. For $j = 0, \dots, n$, let σ^j be the matrix with columns $\sigma_0, \dots, \hat{\sigma}_j, \dots, \sigma_n$. Fix a lattice $L \subseteq \mathbb{Z}^n$, and assume $e = \mathbf{1}$. Then for any $v \in \mathbb{R}^n$, we have the following identity among Dedekind sums:*

$$\sum_{j=0}^n (-1)^j S(L, \sigma^j, \mathbf{1}, v) \Big|_Q = \sum_{j=0}^n (-1)^j S(L \cap \sigma_j^\perp, \sigma^j, \mathbf{1}, v) \Big|_Q. \quad (7)$$

We refer to [5] for proofs of the above statements. Here, in the following two sections, we show how Theorem 1 is applied with a rank 2 example. For simplicity we ignore issues of convergence, and merely remark that all of our manipulations with sums are compatible with the Q -limit process (2).

3.4

Let $L \subset \mathbb{Z}^4$ be the lattice spanned by the first two standard basis vectors, and let

$$\sigma = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad e = \mathbf{1}, \quad \text{and } v = (0, 0, 0, 0).$$

Hence $S(L, \sigma, e, v)$ denotes the absolutely convergent sum

$$\sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{xy(x+2y)^2},$$

where the prime on the summation indicates that we omit the terms (x, y) for which x, y or $x + 2y$ vanish. This sum isn't diagonal, since σ induces 3 different

linear forms on L instead of 2. Note also that the last two rows of σ don't affect the value of the sum; this observation will play an essential role when we apply Proposition 2 to simplify S .

To diagonalize S , we begin with the identity of rational functions

$$\frac{1}{xy(x+2y)^2} = \frac{1}{y(x+2y)^3} + \frac{2}{x(x+2y)^3}. \quad (8)$$

This is true provided none of the denominators vanishes. The numerators of the functions on the right come from the following identity of linear forms on L :

$$\langle w, (1, 2, *, *)^t \rangle = \langle w, 1 \cdot (1, 0, *, *)^t + 2 \cdot (0, 1, *, *)^t \rangle, \quad \text{for all } w \in L. \quad (9)$$

Here the stars denote entries that we don't care about, since they don't affect the value of the linear form on L . We want to sum both sides of (8) over pairs $(x, y) \in \mathbb{Z}^2$ to obtain an identity among Dedekind sums of the form

$$\sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{xy(x+2y)^2} = \sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{y(x+2y)^3} + \sum'_{(x,y) \in \mathbb{Z}^2} \frac{2}{x(x+2y)^3}. \quad (10)$$

However, as written (10) is incorrect. The identity (8) only holds if none of x , y , or $x+2y$ vanish, but the sums on the right of (10) include some of these terms (for instance, the first sum on the right of (10) contains terms (x, y) with $x=0$). We account for this by subtracting two rank 1 Dedekind sums from the right of (10) as “correction terms”:

$$\begin{aligned} \sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{xy(x+2y)^2} &= \sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{y(x+2y)^3} + \sum'_{(x,y) \in \mathbb{Z}^2} \frac{2}{x(x+2y)^3} \\ &\quad - \sum'_{\substack{(x,y) \in \mathbb{Z}^2 \\ x=0}} \frac{1}{y(x+2y)^3} - \sum'_{\substack{(x,y) \in \mathbb{Z}^2 \\ y=0}} \frac{1}{x(x+2y)^3} \\ &= \sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{y(x+2y)^3} + \sum'_{(x,y) \in \mathbb{Z}^2} \frac{2}{x(x+2y)^3} \\ &\quad - \sum'_{y \in \mathbb{Z}} \frac{1}{8y^4} \quad - \sum'_{x \in \mathbb{Z}} \frac{2}{x^4}. \end{aligned} \quad (11)$$

Note that all of the sums on the right of (11) are now diagonal.

This equation is precisely an instance of the reciprocity law (Proposition 2). To see this, apply the law with $\sigma_0, \dots, \sigma_4$ the columns of the matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

and with L and v as above. Note that $\sigma_4 = \sigma_0 + 2\sigma_1$, which is exactly the linear relation (9). The three rank 2 sums (respectively, the two rank 1 sums) are the

left (resp. right) of (7). All other sums vanish identically, either from the linear dependence among the σ_i , or because all terms are meaningless. Notice how we used that the last two rows of the σ_i have no effect on the sum: this enabled us to introduce a linear dependence among the σ_i that killed some of the nondiagonal sums.

To diagonalize a general Dedekind sum $S(L, \sigma, \mathbf{1}, v)$, one considers the configuration $C \subset \mathbb{R}^n$ of linear subspaces consisting of $(L \otimes \mathbb{R})^\perp$ and the spaces generated by the points $\sigma_1, \dots, \sigma_n$. One shows by investigating the geometry of C that a point σ_0 can be found such that when Proposition 2 is applied with the tuple $(\sigma_0, \dots, \sigma_n)$, the resulting Dedekind sums are “closer” to diagonality in a certain sense. It may take several applications of Proposition 2 to express a Dedekind sum as a linear combination of diagonal sums.

3.5

The second rank 2 sum on the right of (11) has index 2. We will show how to make this sum unimodular. Let $\sigma_0, \dots, \sigma_4$ be the columns of the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 2 & 2 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

and let L and v be as above. Then an application of Proposition 2 yields

$$\begin{aligned} \sum'_{(x,y) \in \mathbb{Z}^2} \frac{2}{x(x+2y)^3} &= \sum'_{(x,y) \in \mathbb{Z}^2} \frac{-1}{y(x+2y)^3} + \sum'_{(x,y) \in \mathbb{Z}^2} \frac{1}{x(x+2y)^2 y} \\ &\quad + \sum'_{y \in \mathbb{Z}} \frac{1}{8y^4} + \sum'_{x \in \mathbb{Z}} \frac{2}{x^4}. \end{aligned} \tag{12}$$

Now all the terms on the right of (12) are diagonal and unimodular except for the second rank 2 sum. In fact, this sum is no longer diagonal. However, one further application of Proposition 2 will make this sum diagonal and unimodular. Hence we will have succeeded in expressing the original sum as a finite linear combination of diagonal, unimodular Dedekind sums.

In proceed in general, one must be able to construct the “index-reducing” vector σ_4 as above. An easy argument using Minkowski’s Theorem guarantees the existence of such a vector [1]. To construct this vector in practice, one may use *LLL*-reduction and [4, Conjecture 3.9].

4 Examples

Here we present some numerical examples. For simplicity we compute $\zeta = \zeta_f(\mathfrak{b}, 0)$, where $f = N\mathcal{O}_K$ for various rational integers N , and $\mathfrak{b} = \mathcal{O}_K$. These fields are the first entries in the tables of totally real fields with small discriminant, available from [2].

Cubic fields

- $K = \mathbb{Q}(\theta)$, where $\theta^3 + \theta^2 - 2\theta - 1 = 0$ (discriminant 49).

N	$N \cdot \zeta$	$ N $																	
2	0	4	5	1	2	8	3	11	10	14	16	13	-10	-24	17	-26	19	22	-8
3	2	6	9	8	9	2	12	5	15	14	18	14	18	8	21	6	24	23	

- $K = \mathbb{Q}(\theta)$, where $\theta^3 - 3\theta - 1 = 0$ (discriminant 81).

N	$N \cdot \zeta$	$ N $	$N \cdot \zeta$												
2	0	4	1	7	14	10	-16	13	2	16	-13	19	11	22	72
3	2/3	5	-2	8	7	11	-6	14	0	17	13	20	11	23	62
		6	8/3	9	2/3	12	11/3	15	14/3	18	-64/3	21	14/3	24	29/3

- $K = \mathbb{Q}(\theta)$, where $\theta^3 + \theta^2 - 3\theta - 1 = 0$ (discriminant 148).

N	$N \cdot \zeta$	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	$ N $	$ N \cdot \zeta $	
2	0	4	5	1	-4	7	8	2	3	10	11	-2	-18	13	-20	-22
3	2	6	9	4	9	-10	12	5	15	15	42	18	-32	19	7	82
														20	4	22
														23	12	68

- $K = \mathbb{Q}(\theta)$, where $\theta^3 - \theta^2 - 4\theta - 1 = 0$ (discriminant 169).

N	$N \cdot \zeta$														
2	0	4	3	7	6	10	8	13	2	16	25	19	-238	22	160
3	2	6	8	9	26	12	11	15	17	18	52	21	-10	24	89

- $K = \mathbb{Q}(\theta)$, where $\theta^3 - 4\theta - 1 = 0$ (discriminant 229).

N	$N \cdot \zeta$																
2	0	5	-2	8	7	11	-4	14	24	17	8	20	38	23	202		
3	2	6	0	9	-10	12	18	15	56	18	-16	21	96	24	51		

- $K = \mathbb{Q}(\theta)$, where $\theta^3 - \theta^2 - 4\theta + 3 = 0$ (discriminant 257).

N	$N \cdot \zeta$														
2	0	4	6	7	16	10	4	13	10	16	5	19	42	22	72
3	2	6	8	9	-4	12	8	15	21	18	-16	21	-12	24	58

Quartic fields

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - \theta^3 - 3\theta^2 + \theta + 1 = 0$ (discriminant 725).

N	$N \cdot \zeta$																
2	0	4	5	1	7	-4	10	0	13	-20	16	35	19	-32	22	32	
3	4	6	0	9	-44	12	0	15	0	18	-320	21	-30	24	-84		

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - \theta^3 - 4\theta^2 + 4\theta + 1 = 0$ (discriminant 1125).

N	$N \cdot \zeta$														
2	0	3	4/5	5	4/3	7	124/3	9	-116/5	11	72	13	676/3	15	4

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - 6\theta^2 + 4 = 0$ (discriminant 1600).

N	$N \cdot \zeta$														
2	1	4	1/2	7	4	10	4	13	104	16	45/2	19	224	22	136
3	4	5	4	8	5/2	11	-4	14	-16	17	-84	20	-9	23	60

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - 4\theta^2 - \theta + 1 = 0$ (discriminant 1957).

N	$N \cdot \zeta$										
2	0	3	4	5	20	7	-8	9	52	11	4

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - 5\theta^2 + 5 = 0$ (discriminant 2000).

N	$N \cdot \zeta$	N	$N \cdot \zeta$	N	$N \cdot \zeta$	N	$N \cdot \zeta$	N	$N \cdot \zeta$	N	$N \cdot \zeta$	N	$N \cdot \zeta$		
2	2/5	4	11/5	5	0	7	52	10	0	13	-32	16	31/10	19	412/5
3	4	6	0	9	8	71/10	11	4	14	16	17	296	20	30	568

- $K = \mathbb{Q}(\theta)$, where $\theta^4 - 4\theta^2 + 2 = 0$ (discriminant 2048).

N	$N \cdot \zeta$								
2	1/2	3	4	5	-28	7	8	9	-20

References

1. A. Ash and L. Rudolph, *The modular symbol and continued fractions in higher dimensions*, Invent. Math. **55** (1979), 241–250.
2. J. Buchmann, D. Ford, M. Pohst, M. Olivier, and F. Diaz y Diaz, *Tables of number fields of low degree*, <ftp://megrez.math.u-bordeaux.fr/pub/numberfields/>.
3. D. S. Dummit and D. R. Hayes, *Checking the \mathfrak{p} -adic Stark conjecture when \mathfrak{p} is Archimedean*, Algorithmic number theory (Talence, 1996), Springer, Berlin, 1996, pp. 91–97.
4. P. E. Gunnells, *Computing Hecke eigenvalues below the cohomological dimension*, J. Experimental Math. (to appear), 2000.
5. P. E. Gunnells and R. Sczech, *Evaluation of Dedekind sums, Eisenstein cocycles, and special values of L -functions*, preprint, 1999.
6. D. R. Hayes, *Brumer elements over a real quadratic base field*, Exposition. Math. **8** (1990), no. 2, 137–184.
7. ———, *The partial zeta functions of a real quadratic number field evaluated at $s = 0$* , Number theory (Banff, AB, 1988), de Gruyter, Berlin, 1990, pp. 207–226.

8. ———, *Aligning Brumer-Stark elements into a Hecke character* (working paper, preprint, 1998).
9. R. Sczech, *Eisenstein group cocycles for GL_n and values of L-functions*, Invent. Math. **113** (1993), no. 3, 581–616.
10. C. S. Siegel, *Über die Fourierschen Koeffizienten von Modulformen*, Nachrichten der Akademie der Wissenschaften in Göttingen, Mathematisch-physikalische Klasse (1970), no. 3, 15–56.
11. L. C. Washington, *Introduction to cyclotomic fields*, second ed., Graduate Texts in Mathematics, no. 83, Springer-Verlag, 1997.

Construction of Tables of Quartic Number Fields

Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier

Laboratoire A2X, U.M.R. 5465 du C.N.R.S.
Université Bordeaux I, 351 Cours de la Libération
33405 Talence Cedex, France

Abstract. We explain how to construct efficiently tables of quartic fields by using Dirichlet series coming from Kummer theory, instead of the traditional methods using the geometry of numbers.

1 Introduction

Up to now, methods based on the geometry of numbers have played a leading role in the construction of tables of number fields of fixed degree and signature. We give a brief summary of these methods (see for example [7] for an overview and [9], [11], [12]).

Let L be a number field of degree n , signature (r_1, r_2) with $r_1 + 2r_2 = n$, and discriminant $d(L)$. If L is a primitive number field, a theorem of Hunter asserts that there exists an algebraic integer $\alpha \in \mathbb{Z}_L$ such that $L = \mathbb{Q}(\alpha)$ and for which we have

$$(i) \quad 0 \leq \text{Tr}_{L/\mathbb{Q}}(\alpha) \leq \lfloor n/2 \rfloor$$
$$(ii) \quad \sum_{1 \leq i \leq n} |\alpha_i|^2 \leq \frac{1}{n} (\text{Tr}_{L/\mathbb{Q}}(\alpha))^2 + \gamma_{n-1} \left(\frac{|d(L)|}{n} \right)^{1/(n-1)},$$

where γ_{n-1} is Hermite's constant in dimension $n - 1$ and the α_i s are the conjugates of α in \mathbb{C} . When the field L is imprimitive, we must instead use relative versions of this theorem due to Martinet [10].

Using these bounds, it is straightforward to reduce to the enumeration of a finite number of polynomials defining all the number fields that we are looking for.

The main problem with the above method is that it requires the enumeration of a huge number of polynomials. For each of them we need to check that they are irreducible, to compute the discriminant of the field that they define and then to determine those which give isomorphic number fields. Thus, even for small degrees this method is highly inefficient.

Constructing tables of quadratic fields poses evidently no more problem than detecting squarefree numbers. Thanks to the work of K. Belabas [1], similar constructions are available for cubic fields. We thus concentrate on the next case, the construction of tables of quartic fields.

Such tables have been constructed by Buchmann, Ford and Pohst (see [2] and [3]) using the method described above, and are available for example at the URL

`ftp://megrez.math.u-bordeaux.fr/pub/numberfields/degree4\kern.5em.`
They contain all the quartic fields L with discriminant $|d(L)| \leq 10^6$ for the three possible signatures.

We describe here another method for constructing much more extensive tables of quartic number fields.

In a separate paper (see [5]), we explain how to count number fields L of absolute degree n having a Galois group (of the Galois closure) isomorphic to a given permutation group on n letters G , and discriminant $d(L)$ bounded in absolute value by a given X . This is done essentially by using Kummer theory to compute an explicit formula for the Dirichlet series

$$\Phi_n(G, s) = \sum_{L/\mathbb{Q}} \frac{1}{|d(L)|^s}$$

where the sum is over isomorphism classes of number fields L having Galois group of the Galois closure isomorphic to G . The above paper contains implicitly the construction of the corresponding number fields L , and the aim of the present paper is to make this construction explicit. This leads to the construction of much more extensive tables than is possible using the geometry of numbers, and we could in principle construct tables which are up to 100 or even 1000 times larger than existing tables. We have not yet done so, essentially for storage reasons. It is to be noted that our methods rely in a crucial way on the fact that the group G be solvable, hence we would not be able to construct A_5 or S_5 -extensions for instance.

To illustrate our constructions, we will concentrate on the two specific examples $G = D_4$ (dihedral group with 8 elements) and $G = A_4$. Indeed, the Abelian cases $G = C_2$, $G = C_3$, $G = C_4$, and $G = C_2 \times C_2$ can be treated in a more elementary manner, the case $G = S_3$ is better treated using Belabas's method (see [1]), and the case $G = S_4$ is very similar to the case $G = A_4$, replacing cyclic cubic resolvents by noncyclic cubic resolvents in the construction we give below.

In both the cases $G = D_4$ and $G = A_4$, we will need to study *relative* quadratic extensions, so we start with this first.

2 Relative Quadratic Extensions

Let K be a given base number field, for the moment arbitrary. We will specialize later to the case $[K : \mathbb{Q}] \leq 3$. Denote by (r_1, r_2) the signature of K and by $n = r_1 + 2r_2 = [K : \mathbb{Q}]$ its absolute degree. Consider the following generalization of the Dirichlet series $\Phi_2(C_2, s)$ as follows:

$$\Phi_{2,K}(C_2, s) = \sum_{[L:K]=2} \frac{1}{N(\mathfrak{d}(L/K))^s} .$$

In the above, the sum is over quadratic extensions of K up to K -isomorphism, necessarily with Galois group isomorphic to C_2 , $\mathfrak{d}(L/K)$ is the relative discriminant ideal of L/K , and finally \mathcal{N} denotes the absolute norm from K to \mathbb{Q} .

One of the surprising results of the study made in [4] is that there is a very simple expression for this Dirichlet series, and consequently for the number of quadratic extensions of K of bounded discriminant. Before giving the result, we need a definition.

Definition 1. Let \mathfrak{c} be an integral ideal of K such that $\mathfrak{c} \mid 2\mathbb{Z}_K$.

(1) We define the Selmer group modulo \mathfrak{c}^2 by

$$S_{\mathfrak{c}^2}(K) = \frac{\{u \in K^* / \exists \mathfrak{q}, u\mathbb{Z}_K = \mathfrak{q}^2, (u, \mathfrak{c}) = 1, \exists x, x^2 \equiv u \pmod{*\mathfrak{c}^2}\}}{\{u \in K^{*2} / (u, \mathfrak{c}) = 1\}}.$$

When $\mathfrak{c} = \mathbb{Z}_K$ we will simply speak of the Selmer group of K .

(2) Let α_0 be given such that $\alpha_0 \equiv 1 \pmod{*\mathfrak{c}^2}$. We will denote by $T_{\mathfrak{c}^2}(\alpha_0)$ the set of $\bar{u} \in S_{\mathfrak{c}^2}(K)$ such that for any ideal \mathfrak{c}_1 different from \mathfrak{c} and coprime to α_0 such that $\mathfrak{c} \mid \mathfrak{c}_1 \mid 2\mathbb{Z}_K$, there is no solution to the congruence $x^2 \equiv \alpha_0 u \pmod{*\mathfrak{c}_1^2}$ (it is understood in this definition that \bar{u} is lifted to an element coprime to \mathfrak{c}_1 and not only to \mathfrak{c} , which is always possible).

The following results follows immediately from the results of [4].

Theorem 1. There exists a bijection between quadratic extensions L of K up to K -isomorphism (together with the trivial extension K/K), and the set of triples $(\mathfrak{c}, \mathfrak{a}, \bar{u})$ where \mathfrak{c} is an integral ideal dividing $2\mathbb{Z}_K$, \mathfrak{a} is an integral squarefree ideal coprime to \mathfrak{c} such that the class of \mathfrak{a} belongs to the square of an ideal class in the ray class group $Cl_{\mathfrak{c}^2}(K)$, and $\bar{u} \in T_{\mathfrak{c}^2}(\alpha_0)$, where $\alpha_0 \equiv 1 \pmod{*\mathfrak{c}^2}$ is such that $\mathfrak{a}\mathfrak{q}^2 = \alpha_0\mathbb{Z}_K$ for some ideal \mathfrak{q} . With this notation, the extension corresponding to the triple $(\mathfrak{c}, \mathfrak{a}, \bar{u})$ is $L = K(\sqrt{\alpha_0 u})$ and the relative discriminant $\mathfrak{d}(L/K)$ is equal to $4\mathfrak{a}/\mathfrak{c}^2$.

From the above, it is easy to obtain the following corollary.

Corollary 1. With the above notation, we have

$$\varPhi_{2,K}(C_2, s) = -1 + \frac{1}{2^{2ns}} \sum_{\mathfrak{c} \mid 2} \mathcal{N} \mathfrak{c}^{2s} |S_{\mathfrak{c}^2}(K)| \prod_{\mathfrak{p} \mid \mathfrak{c}} \left(1 - \frac{1}{\mathcal{N} \mathfrak{p}^{2s}}\right) \sum_{\substack{\mathfrak{a} \in Cl_{\mathfrak{c}^2}(K)^2 \\ \mathfrak{a} \text{ squarefree}}} \frac{1}{\mathcal{N} \mathfrak{a}^s}.$$

Furthermore, in [4], the following easy result is proved:

Proposition 1. We have a canonical exact sequence

$$1 \longrightarrow S_{\mathfrak{c}^2}(K) \longrightarrow S_{\mathbb{Z}_K}(K) \longrightarrow \frac{(\mathbb{Z}_K/\mathfrak{c}^2)^*}{(\mathbb{Z}_K/\mathfrak{c})^{*2}} \longrightarrow \frac{Cl_{\mathfrak{c}^2}(K)}{Cl_{\mathfrak{c}^2}(K)^2} \longrightarrow \frac{Cl(K)}{Cl(K)^2} \longrightarrow 1.$$

From this proposition, we immediately deduce the following corollary:

Corollary 2.

$$|S_{\mathfrak{c}^2}(K)| = \frac{2^{r_1+r_2} |Cl_{\mathfrak{c}^2}(K)/Cl_{\mathfrak{c}^2}(K)^2|}{\mathcal{N}\mathfrak{c}} .$$

Putting all this together, a small computation gives (see again [4])

Theorem 2. *Let K be a number field of degree n and signature (r_1, r_2) .*

(1) *We have*

$$\Phi_{2,K}(C_2, s) = -1 + \frac{1}{2^{n(2s-1)+r_2} \zeta_K(2s)} \sum_{\mathfrak{c}|2} \mathcal{N}\mathfrak{c}^{2s-1} \sum_{\chi} \zeta_K(\chi, s) ,$$

where as usual ζ_K denotes the Dedekind zeta function of K , $\zeta_K(\chi, s) = \prod_{\mathfrak{p}} (1 - \chi(\mathfrak{p}) \mathcal{N}\mathfrak{p})^{-s}$ is the L-function attached to the character χ (which we denote in this way so as not to confuse it with the ordinary Dirichlet L-series which will also occur), and in the inner sum χ runs over the quadratic characters of the ray class group $Cl_{\mathfrak{c}^2}(K)$ corresponding to the modulus \mathfrak{c}^2 .

(2) *The number $N_{2,K}(C_2, X)$ of quadratic extensions L of K up to K -isomorphism such that $\mathcal{N}(\mathfrak{d}(L/K)) \leq X$ satisfies*

$$N_{2,K}(C_2, X) \sim \frac{1}{2^{r_2}} \frac{\text{Res}_{s=1} \zeta_K(s)}{\zeta_K(2)} X .$$

We will not need the formula for $N_{2,K}(C_2, X)$ in the present paper, but its simplicity is remarkable, so it deserves to be better known. Although we have proved it using our methods, it can be found slightly hidden in the well known paper of Datskovsky and Wright [8] on relative cubic extensions.

We will directly use Theorem 1 to find all quadratic extensions L/K such that $\mathcal{N}(\mathfrak{d}(L/K)) = N$, as follows.

1) Make the list of all ideals \mathfrak{c} dividing 2 such that $4^n/\mathcal{N}\mathfrak{c}^2 = \mathcal{N}(2/\mathfrak{c})^2$ divides N . For each such ideal \mathfrak{c} , execute the following steps.

2) Compute the elements of $S_{\mathfrak{c}^2}(K)$, and make a list of all integral squarefree ideals \mathfrak{a} which are coprime to \mathfrak{c} and such that $\mathcal{N}\mathfrak{a} = N/\mathcal{N}(2/\mathfrak{c})^2$.

3) For each of the ideals \mathfrak{a} in the list, test whether the class of \mathfrak{a} is a square in the ray class group $Cl_{\mathfrak{c}^2}(K)$. If it is, compute an element α_0 such that $\mathfrak{a}\mathfrak{q}^2 = \alpha_0 \mathbb{Z}_K$ with $\alpha_0 \equiv 1 \pmod{\mathfrak{c}^2}$.

4) For each suitable ideal \mathfrak{a} , compute $T_{\mathfrak{c}^2}(\alpha_0)$ as given in Definition 1 as a subset of $S_{\mathfrak{c}^2}(K)$.

5) The set of extensions L/K such that $\mathcal{N}(\mathfrak{d}(L/K)) = N$ up to K -isomorphism is given by $L = K(\sqrt{\alpha_0 u})$ where α_0 is as above, and $u \in T_{\mathfrak{c}^2}(\alpha_0)$, except that when $\mathfrak{a} = \mathbb{Z}_K$ (in which case one may take $\alpha_0 = 1$), we must exclude $u = 1$ to avoid the trivial extension.

Let us consider an explicit numerical example of the above theorem. Let $K = \mathbb{Q}(\sqrt{-3})$ be the base field, so that $r_2 = 1$ and $n = 2$. The prime 2 is inert in K , hence the only ideals \mathfrak{c} are $\mathfrak{c} = \mathbb{Z}_K$ and $\mathfrak{c} = 2\mathbb{Z}_K$. The ordinary class group is of course trivial, and an easy computation shows that the ray class

group $Cl_{4\mathbb{Z}_K}(K)$ is of order 2, generated by the class of the ideal $\sqrt{-3}\mathbb{Z}_K$. The only nontrivial character of this ray class group is easily seen to be given by $\chi(\mathfrak{a}) = (\frac{-4}{N\mathfrak{a}})$. A short computation gives for $K = \mathbb{Q}(\sqrt{-3})$:

$$\varPhi_{2,K}(C_2, s) = -1 + \frac{1}{2\zeta_K(2s)} \left(\left(1 - \frac{1}{2^{2s}} + \frac{4}{2^{4s}} \right) \zeta_K(s) + \zeta_K \left(\left(\frac{-4}{N(.)} \right), s \right) \right).$$

In addition, we know of course that $\zeta_K(s) = \zeta(s)L((\frac{-3}{.}), s)$.

A small program shows immediately that the first 50 terms of the above Dirichlet series are given by

$$\varPhi_{2,K}(C_2, s) = \frac{2}{13^s} + \frac{1}{16^s} + \frac{2}{21^s} + \frac{1}{25^s} + \frac{2}{37^s} + \frac{2}{48^s} + \frac{1}{49^s} + \cdots$$

This means that, up to K -isomorphism (and *not* up to \mathbb{Q} -isomorphism), there are two quadratic extensions L/K such that $N(\mathfrak{d}(L/K)) = 13$, one such that $N(\mathfrak{d}(L/K)) = 16$, and so on. We have limited to 50 terms for ease of presentation, but on a computer there is no difficulty in obtaining 10^7 terms for example.

Thus, we know that there are exactly 2 quadratic extensions L/K such that $N(\mathfrak{d}(L/K)) = 13$, and we want to compute them explicitly using the above algorithm.

In step 1, we must take $\mathfrak{c} = 2\mathbb{Z}_K$, otherwise $N(2/\mathfrak{c})^2$ is even. In step 2, the list of ideals \mathfrak{a} such that $N\mathfrak{a} = 13$ is evidently given by the two prime ideals above 13, generated by $(7 + \sqrt{-3})/2$ and $(7 - \sqrt{-3})/2$ respectively. An immediate computation shows that the class of both these ideals belongs to $Cl_{\mathfrak{c}^2}^2$, with $\alpha_0 = -1 - 2\sqrt{-3}$ and $\alpha_0 = -1 + 2\sqrt{-3}$ respectively.

Finally, we compute in a naive way that $S_{\mathbb{Z}_K} = \{\pm 1\}$, and since -1 is not a square modulo 4 (in \mathbb{Z}_K), we deduce that the sets $S_{\mathfrak{c}^2}(K)$ and $T_{\mathfrak{c}^2}(\alpha_0)$ are equal to $\{1\}$. Finally, the desired extensions are the two extensions $K(\sqrt{-1 \pm 2\sqrt{-3}})$. Considered as extensions of \mathbb{Q} , these two extensions are \mathbb{Q} -isomorphic to the unique quartic field of discriminant $117 = 3^2 \cdot 13$, with Galois group isomorphic to D_4 , of which an absolute equation is for example $X^4 - X^3 - X^2 + X + 1 = 0$.

Let us do the same computations for $N(\mathfrak{d}(L/K)) = 16$, for which the Dirichlet series tells us that there is only one extension. Since $N = 16$, all possible \mathfrak{c} dividing 2 may be possible, and since 2 is inert in K , we must look at $\mathfrak{c} = \mathbb{Z}_K$ and $\mathfrak{c} = 2\mathbb{Z}_K$.

For $\mathfrak{c} = \mathbb{Z}_K$, we must list ideals of norm 1, hence the only possible ideal \mathfrak{a} is $\mathfrak{a} = \mathbb{Z}_K$, and we can evidently choose $\alpha_0 = 1$. Since $S_{\mathbb{Z}_K}(K) = \{\pm 1\}$, since -1 is not a square modulo $4\mathbb{Z}_K$ and since 1 is excluded in the special case $\mathfrak{a} = \mathbb{Z}_K$, we obtain the extension $L = K(\sqrt{-1})$. Considered as an absolute extension of \mathbb{Q} , this is isomorphic to the unique quartic field of discriminant $144 = 3^2 \cdot 16$ with Galois group isomorphic to $C_2 \times C_2$, of which an absolute equation is for example $X^4 - X^2 + 1 = 0$.

For $\mathfrak{c} = 2\mathbb{Z}_K$, since 2 is inert in K , the only ideal of norm 16 is the ideal $\mathfrak{a} = 4\mathbb{Z}_K$ which is not squarefree, hence no extensions are obtained in this case.

Let us now consider a slightly less easy example. We take $K = \mathbb{Q}(\sqrt{-15})$. This field has class number 2, and in addition the prime 2 is split in K , so the situation is slightly more complicated. We first compute the series $\Phi_{2,K}(C_2, s)$.

For $\mathfrak{c} = \mathbb{Z}_K$, in addition to the trivial character χ_0 , we have the genus character χ_1 which can be defined by the formula

$$\chi_1(\mathfrak{a}) = (-1)^{v_3(\mathcal{N}\mathfrak{a})} \left(\frac{-3}{\mathcal{N}\mathfrak{a}/3^{v_3(\mathcal{N}\mathfrak{a})}} \right) .$$

Indeed, it is immediately checked that this is indeed a character on the ordinary class group (it is multiplicative and trivial on principal ideals) and it is nontrivial since it is equal to -1 on the prime ideal above 5, which is of norm 5. Thus, for $\mathfrak{c} = \mathbb{Z}_K$ the quadratic characters of $Cl_{\mathfrak{c}^2}$ are $\chi = \chi_0$ and $\chi = \chi_1$.

For \mathfrak{c} equal to one of the prime ideals above 2, it is immediately computed that the ray class group is isomorphic to the ordinary class group, so we have the same two characters, except that we must take care that the characters are 0 on \mathfrak{c} .

For $\mathfrak{c} = 2\mathbb{Z}_K$, the ray class group is isomorphic to $C_2 \times C_2$, hence is generated by the character χ_1 and by another character χ_2 . It is easily seen that we may choose χ_2 defined by

$$\chi_2(\mathfrak{a}) = \left(\frac{-4}{\mathcal{N}\mathfrak{a}} \right) .$$

Indeed, this is well defined on the ray class group modulo 4 since, if $\mathfrak{a} = \alpha\mathbb{Z}_K$ with $\alpha \equiv 1 \pmod{4}$ we have $\mathcal{N}\mathfrak{a} = \mathcal{N}(\alpha) \equiv 1 \pmod{4}$, and it is nontrivial since it is equal to -1 on the prime ideal above 3. Thus, the quadratic characters of the ray class group modulo 4 are $\chi = \chi_0, \chi_1, \chi_2$ and $\chi_1\chi_2$.

Summing up, we obtain

$$\begin{aligned} \Phi_{2,K}(C_2, s) &= -1 + \frac{1}{2\zeta_K(2s)} \left(\left(1 - \frac{2}{2^s} + \frac{5}{2^{2s}} - \frac{4}{2^{3s}} + \frac{4}{2^{4s}} \right) \zeta_K(s) \right. \\ &\quad + \left(1 + \frac{2}{2^s} + \frac{5}{2^{2s}} + \frac{4}{2^{3s}} + \frac{4}{2^{4s}} \right) \zeta_K(\chi_1, s) \\ &\quad \left. + \zeta_K \left(\left(\frac{-4}{\mathcal{N}(\cdot)} \right), s \right) + \zeta_K \left(\left(\frac{-4}{\mathcal{N}(\cdot)} \right) \chi_1, s \right) \right) . \end{aligned}$$

A small program shows immediately that the first 50 terms of this Dirichlet series are given by

$$\Phi_{2,K}(C_2, s) = \frac{1}{1^s} + \frac{2}{16^s} + \frac{4}{24^s} + \frac{4}{40^s} + \frac{2}{49^s} + \dots$$

Let us construct the first few corresponding quadratic extensions. First, we compute the sets $S_{\mathfrak{c}^2}(K)$. Denote by \mathfrak{p} and $\bar{\mathfrak{p}}$ the two prime ideals above 2 in K . The possible ideals \mathfrak{c} are $\mathfrak{c} = \mathbb{Z}_K, \mathfrak{p}, \bar{\mathfrak{p}}$ and $2\mathbb{Z}_K$. Since 3 is ramified in K and $S_{\mathbb{Z}_K}(K)$ is of order 4 by Corollary 2, we have

$$S_{\mathbb{Z}_K}(K) = \{\pm 1, \pm 3\} ,$$

and since $-3 \equiv 1 \pmod{4}$ we have

$$S_{\mathfrak{p}^2}(K) = S_{\bar{\mathfrak{p}}^2}(K) = S_{4\mathbb{Z}_K}(K) = \{1, -3\} .$$

The quadratic extension corresponding to the norm $N = 1$ is of course the Hilbert class field of K , clearly obtained with $\mathfrak{c} = 2\mathbb{Z}_K$, $\mathfrak{a} = \mathbb{Z}_K$ and $u = -3$ (since $u = 1$ would give the trivial extension), hence is $K(\sqrt{-3})$, which is well known.

Let us now construct the 2 non- K -isomorphic quadratic extensions L/K such that $N = \mathcal{N}(\mathfrak{d}(L/K)) = 16$. For $\mathfrak{c} = \mathbb{Z}_K$, we must take $\mathfrak{a} = \mathbb{Z}_K$, hence $\alpha_0 = 1$, and hence $T_{\mathbb{Z}_K}(\alpha_0) = \{-1, 3\}$, so we obtain the extensions $L = K(\sqrt{-1})$ and $L = K(\sqrt{3})$. For $\mathfrak{c} = \mathfrak{p}$, $\mathfrak{c} = \bar{\mathfrak{p}}$, or $\mathfrak{c} = 2\mathbb{Z}_K$, it is immediately seen that all integral ideals \mathfrak{a} having a suitable norm are either not squarefree or not coprime to \mathfrak{c} , so there are no other extensions, as predicted by the Dirichlet series.

To finish this example, we construct the 4 non- K -isomorphic quadratic extensions L/K such that $N = \mathcal{N}(\mathfrak{d}(L/K)) = 24$. Since $16 \nmid N$, we can take $\mathfrak{c} = \mathfrak{p}$, $\mathfrak{c} = \bar{\mathfrak{p}}$ and $\mathfrak{c} = 2\mathbb{Z}_K$. Consider first $\mathfrak{c} = \mathfrak{p}$. We must find the list of squarefree ideals prime to \mathfrak{c} of norm 6, and clearly there is only one such ideal, the ideal $\mathfrak{a} = \bar{\mathfrak{p}}\mathfrak{p}_3$, where \mathfrak{p}_3 denotes the unique prime ideal above 3. We check immediately that \mathfrak{a} belongs to the square of an ideal class in $Cl_{\mathfrak{c}^2}(K)$ and that we can take $\alpha_0 = (3 + \sqrt{-15})/2$ (or its conjugate, depending on the specific choice of \mathfrak{p}). The only possible ideal \mathfrak{c}_1 that we have to check is $\mathfrak{c}_1 = 2\mathbb{Z}_K$, which is not coprime to α_0 . Thus $T_{\mathfrak{c}^2}(\alpha_0) = \{1, -3\}$ giving the two extensions $L = K(\sqrt{(3 + \sqrt{-15})/2})$ and $L = K(\sqrt{(-9 - 3\sqrt{-15})/2})$. The choice $\mathfrak{c} = \bar{\mathfrak{p}}$ would give the two other extensions $L = K(\sqrt{(3 - \sqrt{-15})/2})$ and $L = K(\sqrt{(-9 + 3\sqrt{-15})/2})$. Finally, for $\mathfrak{c} = 2\mathbb{Z}_K$ we would need an ideal of norm 24, which would not be coprime to \mathfrak{c} , so no extensions are obtained in this case.

In a similar manner, we can easily consider examples where 2 is ramified and the class group is nontrivial (for example $K = \mathbb{Q}(\sqrt{-20})$), or other base fields K such as cyclic cubic fields. Some new phenomena appear in these cases, but nothing essential, and we leave this as practice for the reader.

To finish this section, we note that even though in practice it is very easy to compute the groups $S_{\mathfrak{c}^2}(K)$ by simple enumeration of the suitable elements of $S_{\mathbb{Z}_K}(K)$, the Selmer group of K which is generated by the units and the virtual units (see [7]), there is a completely algorithmic way of doing it by using directly the exact sequence of Proposition 1.

3 Constructing D_4 -Extensions

The first application of the above results and constructions is to the computation of (extensive) tables of quartic D_4 -extensions of \mathbb{Q} . We have for example computed that the *number* of such extensions of absolute discriminant less than or equal to 10^{14} is equal to 5232538688240. The computation took less than 4 days

of CPU time. It is evidently out of the question to make a table of the corresponding polynomials, and in practice we have constructed such a table only up to 10^7 , because of storage and not of time considerations. We could if necessary construct a table up to 10^9 with approximately 3GB of storage and 48 hours of CPU time. This is 1000 times further than the published tables obtained by the geometry of numbers. With present day computers, it is quite plausible that using the geometry of numbers, we could construct tables which go 10 times further than before, but it seems unlikely that we can go 1000 times further, and certainly not in 48 hours of CPU time.

To construct D_4 -extensions L/\mathbb{Q} , we note first that such extensions are *imprimitive*, in other words that there exists a quadratic field K such that $K \subset L$, so that L is a quadratic extension of K . Furthermore, it is clear that if τ is the unique nontrivial element of the Galois group of K/\mathbb{Q} then L and $\tau(L)$ are non- K -isomorphic but \mathbb{Q} -isomorphic quadratic extensions of K . Finally, note the formula $|d(L)| = d(K)^2 \mathcal{N}(\mathfrak{d}(L/K))$.

Of course, not all imprimitive quartic extensions of \mathbb{Q} are D_4 -extensions. They can also be C_4 -extensions or $C_2 \times C_2$ -extensions. However these extensions are much rarer (there are approximately $0.0523 X D_4$ -extensions of absolute discriminant up to X , compared to $0.122 X^{1/2} C_4$ -extensions and $0.00275 X^{1/2} \log^2 X C_2 \times C_2$ -extensions, see [5]), are easy to detect (for example, $C_2 \times C_2$ extensions are the only imprimitive quartic fields with square discriminant) and are much easier to construct. Thus, constructing tables of D_4 -extensions is essentially equivalent to constructing tables of imprimitive quartic fields.

To construct a table of D_4 -extensions of absolute discriminant up to X , we thus proceed as follows.

1) First construct a table of quadratic fields K such that $|d(K)| \leq \sqrt{X}$. This essentially amounts to finding squarefree numbers, and in addition since \sqrt{X} will be small (31622 for $X = 10^9$), this computation is immediate. For each such quadratic field, compute the class group, the ideals \mathfrak{c} dividing 2, the ray class groups $Cl_{\mathfrak{c}^2}(K)$ and the groups $S_{\mathfrak{c}^2}(K)$ (of course this is not stored, but done on the fly as we go through each quadratic field in sequence).

2) For each quadratic field K , use the techniques explained in Section 2 to construct all quadratic extensions L of K such that $\mathcal{N}(\mathfrak{d}(L/K)) \leq B = \lfloor X/d(K)^2 \rfloor$. More precisely, compute the list \mathcal{L} of all integral squarefree ideals \mathfrak{a} of norm less than or equal to B whose class is a square in the ordinary class group. For each ideal $\mathfrak{c} \mid 2$, extract from \mathcal{L} those ideals \mathfrak{a} coprime to \mathfrak{c} whose class is a square in the ray class group modulo \mathfrak{c}^2 (this is easily done once we know that the class of \mathfrak{a} is a square in the ordinary class group) of norm less than or equal to $B/\mathcal{N}(2/\mathfrak{c})^2$. For each such ideal \mathfrak{a} compute $\alpha_0 \equiv 1 \pmod{\mathfrak{c}^2}$ such that $\mathfrak{a}\mathfrak{q}^2 = \alpha_0 \mathbb{Z}_K$ for some ideal \mathfrak{q} . The desired extensions are $K(\sqrt{\alpha_0 u})$ for $\overline{u} \in T_{\mathfrak{c}^2}(\alpha_0)$, excluding the trivial extension.

3) To avoid \mathbb{Q} -isomorphic fields, in the above construction we identify a triple $(\mathfrak{c}, \mathfrak{a}, \overline{u})$ with its conjugate by the nontrivial Galois automorphism of K . To avoid $C_2 \times C_2$ -extensions, we exclude ideals \mathfrak{a} of square norm, and to avoid C_4 -

extensions, we exclude ideals \mathfrak{a} of norm equal to a square times the discriminant of K (this occurs only when K is a real quadratic field).

4 Constructing A_4 -Extensions

The situation is considerably more complicated in this case, but it is still possible to construct extensive tables of A_4 -extensions. In fact, since according to [4] the number of such extensions with absolute discriminant up to X is approximately equal to $0.0179 X^{1/2} \log X$, it is possible to go much further than the typical bound 10^9 we gave for D_4 -extensions. For example, we have computed that the number of such extensions up to 10^{11} is equal to 104766 in less than 24 hours of CPU time, and using the method given in this section, it would be easy to build the corresponding table in essentially the same amount of time.

Let L/\mathbb{Q} be a quartic extension, let N be its Galois closure in \mathbb{C} , and assume that the Galois group of N/\mathbb{Q} is isomorphic to A_4 . The field N has a unique cubic subfield K_3 which is cyclic over \mathbb{Q} . The extension N/K_3 is a biquadratic extension, hence contains three quadratic subextensions L_i/K_3 for $0 \leq i \leq 2$, and these subextensions have trivial norm, in other words $L_i = K_3(\sqrt{\alpha_i})$ with $N_{K_3/\mathbb{Q}}(\alpha_i)$ a square of \mathbb{Q} . Finally the discriminant of L is given by $d(L) = d(K_3) N_{K_3/\mathbb{Q}}(\mathfrak{d}(L_i/K_3))$ for any i .

Denote by σ a generator of $\text{Gal}(N/L)$, which can also be considered as a generator of $\text{Gal}(K_3/\mathbb{Q})$. We can set $\alpha = \alpha_0$ and take $\alpha_i = \sigma^i(\alpha)$. If θ is a square root of α in N , we have $L = \mathbb{Q}(\eta)$ with

$$\eta = \theta + \sigma(\theta) + \sigma^2(\theta) = \text{Tr}_{N/L}(\theta) .$$

Explicitly, if $X^3 + aX^2 + bX + c$ is the minimal polynomial of α , a defining equation for L/\mathbb{Q} is given for example by the polynomial

$$P_\alpha(X) = X^4 - 2(a^2 - 2b)X^2 + 8cX + (a^4 - 4a^2b + 8ac)$$

whose discriminant is equal to $2^{12}(c - ab)^2$ times that of the polynomial $X^3 + aX^2 + bX + c$.

Enumerating cyclic cubic fields is very easy, so there remains to explain how to enumerate quadratic extensions L_i/K_3 with trivial norm. This is done by generalizing the definitions and results given in Section 2 when there are no restrictions.

Definition 2. Let K be a number field, \mathfrak{c} an ideal of K dividing 2, and denote by \mathcal{N} the norm from K to \mathbb{Q} .

(1) We define the square ray class group modulo \mathfrak{c}^2 by

$$Cl_{\mathfrak{c}^2}[\mathcal{N}] = \frac{\{\mathfrak{a}/ (\mathfrak{a}, \mathfrak{c}) = \mathbb{Z}_K, \mathcal{N}(\mathfrak{a}) \text{ square}\}}{\{\beta \mathbb{Z}_K / \beta \equiv 1 \pmod{\mathfrak{c}^2}, \mathcal{N}(\beta) \text{ square}\}} .$$

When $\mathfrak{c} = \mathbb{Z}_K$, we will simply speak of the square class group.

(2) We define the square Selmer group modulo \mathfrak{c}^2 by

$$S_{\mathfrak{c}^2}[\mathcal{N}] = \{\overline{u} \in S_{\mathfrak{c}^2}(K) / \mathcal{N}(u) \text{ square}\} .$$

(3) Let α_0 be given such that $\alpha_0 \equiv 1 \pmod{*\mathfrak{c}^2}$ and $\mathcal{N}(\alpha_0)$ a square. We will denote by $T_{\mathfrak{c}^2}(\alpha_0)[\mathcal{N}]$ the set of $\overline{u} \in S_{\mathfrak{c}^2}[\mathcal{N}]$ such that for any ideal \mathfrak{c}_1 different from \mathfrak{c} and coprime to α_0 such that $\mathfrak{c} \mid \mathfrak{c}_1 \mid 2\mathbb{Z}_K$, there is no solution to the congruence $x^2 \equiv \alpha_0 u \pmod{*\mathfrak{c}_1^2}$. In other words,

$$T_{\mathfrak{c}^2}(\alpha_0)[\mathcal{N}] = T_{\mathfrak{c}^2}(\alpha_0) \cap S_{\mathfrak{c}^2}[\mathcal{N}] .$$

The analog of Theorem 1 is then as follows.

Theorem 3. Let K be a number field. There exists a bijection between quadratic extensions L of K with trivial norm up to K -isomorphism (together with the trivial extension K/K), and the set of triples $(\mathfrak{c}, \mathfrak{a}, \overline{u})$ where \mathfrak{c} is an integral ideal dividing $2\mathbb{Z}_K$, \mathfrak{a} is an integral squarefree ideal coprime to \mathfrak{c} which is of square norm, and such that the class of \mathfrak{a} belongs to the square of an ideal class in the square ray class group $Cl_{\mathfrak{c}^2}[\mathcal{N}]$, and $\overline{u} \in T_{\mathfrak{c}^2}(\alpha_0)[\mathcal{N}]$, where $\alpha_0 \equiv 1 \pmod{*\mathfrak{c}^2}$ of square norm is such that $\mathfrak{a}\mathfrak{q}^2 = \alpha_0\mathbb{Z}_K$ for some ideal \mathfrak{q} . With this notation, the extension corresponding to the triple $(\mathfrak{c}, \mathfrak{a}, \overline{u})$ is $L = K(\sqrt{\alpha_0\overline{u}})$ and the relative discriminant $\mathfrak{d}(L/K)$ is equal to $4\mathfrak{a}/\mathfrak{c}^2$.

From this theorem it is easy to deduce the following analog of Corollary 1.

Corollary 3. Denote by $\Phi_{2,K}(C_2, s)[\mathcal{N}]$ the Dirichlet series which is the generating function of quadratic extensions of square norm. Then

$$\Phi_{2,K}(C_2, s)[\mathcal{N}] = -1 + \frac{1}{2^{2ns}} \sum_{\mathfrak{c}|2} \mathcal{N}\mathfrak{c}^{2s} |S_{\mathfrak{c}^2}[\mathcal{N}]| \prod_{\mathfrak{p}|\mathfrak{c}} \left(1 - \frac{1}{\mathcal{N}\mathfrak{p}^{2s}}\right) \sum_{\substack{\mathfrak{a} \in Cl_{\mathfrak{c}^2}[\mathcal{N}]^2 \\ \mathfrak{a} \text{ squarefree}}} \frac{1}{\mathcal{N}\mathfrak{a}^s} .$$

Note that in the above sum the ideals \mathfrak{a} are of square norm, hence the Dirichlet series $\Phi_{2,K}(C_2, s)[\mathcal{N}]$ is a Dirichlet series in the variable $2s$.

The analog of Corollary 2 is the following, in the case we are interested in (see [4] for the general case).

Proposition 2. Let K_3 be a cyclic cubic field, and set $x(\mathfrak{c}) = 2$ if $\mathfrak{c} = 2\mathbb{Z}_{K_3}$, $x(\mathfrak{c}) = 1$ otherwise. Then

$$S_{\mathfrak{c}^2}[\mathcal{N}] = \frac{2^2 x(\mathfrak{c}) |Cl_{\mathfrak{c}^2}[\mathcal{N}] / Cl_{\mathfrak{c}^2}[\mathcal{N}]^2|}{\mathcal{N}\mathfrak{c}} .$$

Putting everything together, a small computation gives (see [4])

Theorem 4. Let K_3 be a cyclic cubic field. Then

$$\begin{aligned} \Phi_{2,K_3}(C_2, s)[\mathcal{N}] = -1 + \frac{1}{2^{6s-2}} \sum_{\mathfrak{c}|2} \mathcal{N}\mathfrak{c}^{2s-1} x(\mathfrak{c}) \prod_{\mathfrak{p}|\mathfrak{c}} \left(1 - \frac{1}{\mathcal{N}\mathfrak{p}^{2s}}\right) \\ \cdot \sum_{\chi \in \widehat{Cl_{\mathfrak{c}^2}[\mathcal{N}] / Cl_{\mathfrak{c}^2}[\mathcal{N}]^2}} F(\chi, s) , \end{aligned}$$

where

$$F(\chi, s) = \prod_{p \in \mathbb{Z}_{K_3} = \mathfrak{p}_1 \mathfrak{p}_2 \mathfrak{p}_3} \left(1 + \frac{\chi(\mathfrak{p}_1 \mathfrak{p}_2) + \chi(\mathfrak{p}_1 \mathfrak{p}_3) + \chi(\mathfrak{p}_2 \mathfrak{p}_3)}{p^{2s}} \right).$$

By summing over all cyclic cubic fields, we obtain

Corollary 4. *With the notation of the above theorem, we have*

$$\Phi_4(A_4, s) = \frac{1}{3} \sum_{K_3/\mathbb{Q}} \frac{1}{f(K_3)^{2s}} \Phi_{2, K_3}(C_2, s)[\mathcal{N}] ,$$

where the sum is over all isomorphism classes of cyclic cubic fields K_3/\mathbb{Q} .

Here, we simply want to use Theorem 3 for the construction of tables of A_4 -extensions. As already noted, essentially the same method leads to the construction of tables of S_4 -extensions, simply by replacing cyclic cubic fields by noncyclic ones.

To construct a table of quartic A_4 -extensions of \mathbb{Q} of discriminant up to X , we thus proceed as follows.

1) First construct a table of cyclic cubic fields K_3 such that $|f(K_3)| \leq \sqrt{X}$, where $f(K_3) = \sqrt{d(K_3)}$ is the conductor of K_3 . This is very easily done since the structure of set of isomorphism classes of cyclic cubic fields is completely known (see for example [6]), and is very fast since we need to consider conductors only up to the square root of X . Then for each such cyclic cubic field, compute the square class group, the ideals \mathfrak{c} dividing 2, the square ray class groups and the square Selmer groups modulo \mathfrak{c}^2 . Compute also explicitly the action of a generator σ of the Galois group $\text{Gal}(K_3/\mathbb{Q})$ on elements and ideals of K_3 (see [6]). Then for each such K_3 perform the following steps.

2) Make a list of squarefree integral ideals \mathfrak{b} of norm less than or equal to $B = \sqrt{X/f(K_3)^2} = \sqrt{X/d(K_3)}$ such that \mathfrak{b} is divisible only by prime ideals above primes of \mathbb{Q} which are split in K_3 , and such that \mathfrak{b} is not divisible by two distinct prime ideals above the same split prime of \mathbb{Q} . Let \mathcal{L} be the list of ideals \mathfrak{a} of the form $\mathfrak{a} = \mathfrak{b}\sigma(\mathfrak{b})$ (where \mathfrak{b} ranges in the list that we have just found) and whose class is a square in the square class group. For each ideal $\mathfrak{c} \mid 2$, extract from \mathcal{L} those ideals \mathfrak{a} coprime to \mathfrak{c} whose class is a square in the square class group modulo \mathfrak{c}^2 (once again, this is now easily done). For each such ideal \mathfrak{a} compute $\alpha_0 \equiv 1 \pmod{\mathfrak{c}^2}$ of square norm such that $\mathfrak{a}\mathfrak{q}^2 = \alpha_0 \mathbb{Z}_{K_3}$ for some ideal \mathfrak{q} .

3) The quadratic extensions L_i of trivial norm of K_3 with $\mathcal{N}(L_i/K_3) \leq X/d(K_3)$ are the $K_3(\sqrt{\alpha_0 u})$ for $\bar{u} \in T_{\mathfrak{c}^2}(\alpha_0)[\mathcal{N}]$, excluding the trivial extension. To avoid \mathbb{Q} -isomorphic fields, in the above construction we must identify a triple $(\mathfrak{c}, \mathfrak{a}, \bar{u})$ with its two Galois conjugates by σ and σ^2 . The corresponding quartic A_4 -extensions L of \mathbb{Q} are obtained by computing the equation of the cubic polynomial satisfied by $\alpha_0 u$ over \mathbb{Q} and applying the formula given above for the corresponding quartic.

References

1. K. Belabas, A fast algorithm to compute cubic fields, *Math. Comp.* **66** (1997), 1213–1237.
2. J. Buchmann and D. Ford, On the computation of totally real quartic fields of small discriminant, *Math. Comp.* **52** (1989), 161–174.
3. J. Buchmann, D. Ford, and M. Pohst, Enumeration of quartic fields of small discriminant, *Math. Comp.* **61** (1993), 873–879.
4. H. Cohen, F. Diaz y Diaz and M. Olivier, Density of number field discriminants, in preparation.
5. H. Cohen, F. Diaz y Diaz and M. Olivier, Counting discriminants of number fields, this volume.
6. H. Cohen, A course in computational algebraic number theory (third printing), GTM **138**, Springer-Verlag, 1996.
7. H. Cohen, Advanced topics in computational number theory, GTM **193**, Springer-Verlag, 2000.
8. B. Datskovsky and D. J. Wright, Density of discriminants of cubic extensions, *J. Reine Angew. Math.* **386** (1988), 116–138.
9. P. Letard, Construction de corps de nombres de degré 7 et 9, Thesis, Université Bordeaux I (1995).
10. J. Martinet, Méthodes géométriques dans la recherche des petits discriminants, *Prog. Math.* **59**, Birkhäuser, Boston (1985), 147–179.
11. M. Pohst, On the computation of number fields of small discriminants including the minimum discriminants of sixth degree fields, *J. Number Theory* **14** (1982), 99–117.
12. A. Schwarz, M. Pohst, and F. Diaz y Diaz, A table of quintic number fields, *Math. Comp.* **63** (1994), 361–376.

Counting Discriminants of Number Fields of Degree up to Four

Henri Cohen, Francisco Diaz y Diaz, and Michel Olivier

Laboratoire A2X, U.M.R. 5465 du C.N.R.S.
Université Bordeaux I, 351 Cours de la Libération
33405 Talence Cedex, France

Abstract. For each permutation group G on n letters with $n \leq 4$, we give results, conjectures and numerical computations on discriminants of number fields L of degree n over \mathbb{Q} such that the Galois group of the Galois closure of L is isomorphic to G .

1 Introduction

The aim of this paper is to regroup results and conjectures on discriminant counts of number fields of degree less than or equal to 4, from a theoretical, practical, and numerical point of view. Proofs will be given in a forthcoming paper.

We only consider absolute number fields, and for simplicity we do not distinguish between different signatures, although this can easily be done. We denote by G the Galois group of the Galois closure.

If G is a permutation group on n letters, we write

$$\Phi_n(G, s) = \sum_{L/\mathbb{Q}} \frac{1}{|d(L)|^s} \quad \text{and} \quad N_n(G, X) = \sum_{L/\mathbb{Q}, |d(L)| \leq X} 1 ,$$

where in both cases the summation is over isomorphism classes of number fields L of degree n over \mathbb{Q} such that the Galois group of the Galois closure of L is isomorphic to G and $d(L)$ denotes the absolute discriminant of L .

Important Remark. Certain authors, in particular Datskowsky, Wright and Yukie (see [7], [14], [15]) count number fields in a fixed algebraic closure of \mathbb{Q} , which is perhaps more natural. This is the same as $N_n(G, X)$ when G is of cardinality equal to n , i.e., when the extensions L are Galois. Otherwise, in the range of our study ($n \leq 4$), their count is equal to $m(G)N_n(G, X)$, where $m(S_3) = 3$, $m(D_4) = 2$, $m(A_4) = m(S_4) = 4$.

For each group G , we give the results in the following form: we first give expressions for $\Phi_n(G, s)$ which are as explicit as possible. Then we give an asymptotic formula for $N_n(G, X)$ which is usually directly deduced from the formula for $\Phi_n(G, s)$. In some cases the asymptotic formula can be refined, but usually only conjecturally. We then explain the method that we have used to compute $N_n(G, X)$ exactly. Finally, we give a table of $N_n(G, 10^k)$ for increasing values of

k as well as a comparison of this data with the most refined result or conjecture on the asymptotic behavior of $N_n(G, X)$. The upper bound chosen for k depends on the time and space necessary to compute the data: we should not need more than one week of CPU time and 1GB of RAM.

It should be stressed that although we only give the *number* $N_n(G, X)$ of suitable fields, the same methods can also be used to compute explicitly a defining equation for these number fields, but the storage problem makes this impractical for more than a few million fields.

2 Degree 2 Fields with $G \simeq C_2$

2.1 Dirichlet Series and Asymptotic Formulas

Using the characterization of a fundamental quadratic discriminant, it is easy to show that

$$\begin{aligned}\Phi_2(C_2, s) &= \left(1 + \frac{1}{2^{2s}} + \frac{2}{2^{3s}}\right) \prod_{p \equiv 1 \pmod{2}} \left(1 + \frac{1}{p^s}\right) - 1 \\ &= \left(1 - \frac{1}{2^s} + \frac{2}{2^{2s}}\right) \prod_p \left(1 + \frac{1}{p^s}\right) - 1 = \left(1 - \frac{1}{2^s} + \frac{2}{2^{2s}}\right) \frac{\zeta(s)}{\zeta(2s)} - 1.\end{aligned}$$

From the above formulas, we easily deduce a crude form of the asymptotic formula:

$$\begin{aligned}N_2(C_2, X) &\sim c(C_2) X \quad \text{with} \\ c(C_2) &= \frac{1}{\zeta(2)} = \frac{6}{\pi^2} = 0.607927101854026628663276779\dots\end{aligned}$$

It is known that we have the more precise result

$$N_2(C_2, X) = c(C_2) X + O(X^{1/2} \exp(-c \log X^{3/5} \log \log X^{-1/5}))$$

for some positive constant c , and under the Riemann Hypothesis, that

$$N_2(C_2, X) = c(C_2) X + O(X^\alpha)$$

for any $\alpha > 8/25$ (see for example [13], Notes du Chapitre I.3). It is conjectured, and this is strongly confirmed by the tables, that we can take any $\alpha > 1/4$ in the error term.

2.2 Numerical Computation

Since $1/\zeta(2s) = \sum_{m \geq 1} \mu(m)/m^{2s}$, it is easy to deduce from the formula for $\Phi_2(C_2, s)$ the formula

$$\begin{aligned}N_2(C_2, X) &= -1 + \sum_{1 \leq m \leq \sqrt{X/4}} \mu(m) \left(\left\lfloor \frac{X+m^2}{2m^2} \right\rfloor + 2 \left\lfloor \frac{X}{4m^2} \right\rfloor \right) \\ &\quad + 2 \sum_{\sqrt{X/4} < m \leq \sqrt{X/3}} \mu(m) + \sum_{\sqrt{X/3} < m \leq \sqrt{X}} \mu(m).\end{aligned}$$

This is the formula that we have used in exact computations. Note that, although we could use directly the Dirichlet series $\varPhi_2(C_2, s)$, this would be much less efficient.

2.3 Table

In this table, we let $P_2(C_2, X) = \lfloor c(C_2) X \rfloor$ be the predicted value and $E_2(C_2, X) = (N_2(C_2, X) - P_2(C_2, X))/X^{1/4}$ rounded to 5 decimals.

X	$N_2(C_2, X)$	$P_2(C_2, X)$	$E_2(C_2, X)$
10^1	6	6	0
10^2	61	61	0
10^3	607	608	-0.17783
10^4	6086	6079	0.70000
10^5	60786	60793	-0.39364
10^6	607925	607927	-0.06325
10^7	6079285	6079271	0.24896
10^8	60792709	60792710	-0.01000
10^9	607927069	607927102	-0.18557
10^{10}	6079270822	6079271019	0.62297
10^{11}	60792710200	60792710185	0.02667
10^{12}	607927101751	607927101854	-0.10300
10^{13}	6079271018463	6079271018540	-0.04330
10^{14}	60792710186342	60792710185403	0.29694
10^{15}	607927101852652	607927101854027	-0.24451
10^{16}	6079271018544414	6079271018540266	0.41480
10^{17}	60792710185393816	60792710185402663	-0.49750
10^{18}	607927101854026495	607927101854026629	-0.00424
10^{19}	6079271018540242468	6079271018540266287	-0.42357

A notable feature of this table, common to most of the tables that we give, is the changes in sign of the error term, showing that there is no systematic bias. Thus, only the order of magnitude of the error term can be questioned, but the existence of an additional main term seems unlikely.

3 Degree 3 Fields with $G \simeq C_3$

3.1 Dirichlet Series and Asymptotic Formulas

From the characterization of discriminants of cyclic cubic fields (see for example [5], Section 6.4.2), it is easy to show that

$$\varPhi_3(C_3, s) = \frac{1}{2} \left(1 + \frac{2}{3^{4s}} \right) \prod_{p \equiv 1 \pmod{6}} \left(1 + \frac{2}{p^{2s}} \right) - \frac{1}{2} .$$

From the above, we deduce a crude form of the asymptotic formula:

$$\begin{aligned} N_3(C_3, X) &\sim c(C_3) X^{1/2} \quad \text{with} \\ c(C_3) &= \frac{11\sqrt{3}}{36\pi} \prod_{p \equiv 1 \pmod{6}} \left(1 - \frac{2}{p(p+1)}\right) \\ &= 0.1585282583961420602835078203575\dots \end{aligned}$$

It is easy to refine this to the more precise result

$$N_3(C_3, X) = c(C_3) X^{1/2} + O(X^\alpha)$$

for $\alpha = 1/3$, and probably with some effort for some $\alpha < 1/3$.

In view of the positions of the poles of the function $\Phi_3(C_3, s)$ it is reasonable to conjecture that we can take $\alpha = 1/6 + \varepsilon$ for all $\varepsilon > 0$.

3.2 Numerical Computation

From the explicit formula for $\Phi_3(C_3, s)$ or equivalently from the explicit description of cyclic cubic fields, it is easy to deduce the formula

$$N_3(C_3, X) = -\frac{1}{2} + 3 \sum_{1 \leq m \leq \sqrt{X}/9} f_6(m) 2^{\omega(m)-1} + \sum_{\sqrt{X}/9 < m \leq \sqrt{X}} f_6(m) 2^{\omega(m)-1},$$

where $f_6(m) = 1$ if m is equal to a squarefree product of primes congruent to 1 modulo 6, and to 0 otherwise, and $\omega(m)$ is the usual function counting the number of prime divisors of m .

This is the formula that we have used in exact computations. As for the C_2 case, it would be much less efficient to use directly the Dirichlet series.

3.3 Table

In this table, we let $P_3(C_3, X) = \lfloor c(C_3) X^{1/2} \rfloor$ be the predicted value and $E_3(C_3, X) = (N_3(C_3, X) - P_3(C_3, X))/X^{1/6}$ rounded to 5 decimals.

X	$N_3(C_3, X)$	$P_3(C_3, X)$	$E_3(C_3, X)$
10^1	0	1	-0.68129
10^2	2	2	0
10^3	5	5	0
10^4	16	16	0
10^5	51	50	0.14678
10^6	159	159	0
10^7	501	501	0
10^8	1592	1585	0.32491
10^9	5008	5013	-0.15811
10^{10}	15851	15853	-0.04309
10^{11}	50152	50131	0.30824
10^{12}	158542	158528	0.14000
10^{13}	501306	501310	-0.02725
10^{14}	1585249	1585283	-0.15781
10^{15}	5013206	5013104	0.32255
10^{16}	15852618	15852826	-0.44812
10^{17}	50131008	50131037	-0.04257
10^{18}	158528150	158528258	-0.10800
10^{19}	501309943	501310370	-0.29091

4 Degree 3 Fields with $G \simeq S_3 \simeq D_3$

4.1 Dirichlet Series and Asymptotic Formulas

In this case, we may use methods coming from Kummer theory to compute the Dirichlet series and to deduce an asymptotic estimate for $N_3(S_3, X)$, but the results are too complicated to state here.

On the other hand, we have the celebrated Davenport-Heilbronn theorem (see [8], [9] and [6], Chapter 8) which asserts that

$$N_3(S_3, X) \sim c(S_3) X \text{ with}$$

$$c(S_3) = \frac{1}{3\zeta(3)} = 0.27730245752690248956104209294\dots$$

From the work of K. Belabas (see [1]), it is known that we can refine this estimate to

$$N_3(S_3, X) = c(S_3) X + O(X \exp(-c\sqrt{\log X \log \log X}))$$

for any $c < 1/24$.

However, much more is conjectured to be true. From the work of Shintani and heuristics of D. Roberts (see [11], [12], [10], where the constant $\zeta(2)$ must be omitted), it is believed that there is an additional main term and that we have in fact

$$N_3(S_3, X) = c(S_3) X + c'(S_3) X^{5/6} - \frac{c(C_3)}{3} X^{1/2} + o(X^{1/2})$$

with

$$\begin{aligned} c'(S_3) &= \frac{3(3 + \sqrt{3})\Gamma(1/3)^3}{10\pi^3} \frac{\zeta(1/3)}{\zeta(5/3)} \\ &= -0.40348363666394679863364025671534\dots \end{aligned}$$

4.2 Numerical Computation

We refer to the work of K. Belabas ([2] and [6], Chapter 8) for the use of the Davenport-Heilbronn method to compute $N_3(S_3, X)$. The details would be too long to state here. We have simply copied Belabas's results. Note that to obtain the table below from his, one must first add the contributions of the complex and totally real cubic fields, since we do not distinguish between different signatures, and then subtract the contribution of cyclic cubic fields given in the table above.

We could also use the approach based on Kummer theory. This would certainly be much less efficient since Belabas's method gives cubic fields in essentially linear time. The most serious obstruction would not be so much the complexity of the formula or the ray class groups which occur, but the fact that we must sum over all quadratic discriminants up to X .

4.3 Table

In this table, we let $P_3(S_3, X) = \left[c(S_3)X + c'(S_3)X^{5/6} - \frac{c(C_3)}{3}X^{1/2} \right]$ be the predicted value using the refined heuristics, and $E_3(S_3, X) = (N_3(S_3, X) - P_3(S_3, X))/X^{1/2}$ rounded to 5 decimals.

X	$N_3(S_3, X)$	$P_3(S_3, X)$	$E_3(S_3, X)$
10^1	0	0	0
10^2	7	8	-0.10000
10^3	149	148	0.03162
10^4	1886	1898	-0.12000
10^5	21794	21791	0.00949
10^6	236858	236901	-0.04300
10^7	2497935	2497967	-0.01012
10^8	25855883	25856912	-0.10290
10^9	264539133	264541514	-0.07529
10^{10}	2686092328	2686091377	0.00951
10^{11}	27138004413	27137996056	0.02643

5 Degree 4 Fields with $G \simeq C_4$

5.1 Dirichlet Series and Asymptotic Formulas

By studying discriminants of cyclic quartic extensions, it is not difficult to show that

$$\begin{aligned}\Phi_4(C_4, s) &= \frac{\zeta(2s)}{2\zeta(4s)} \left(\left(1 - \frac{1}{2^{2s}} + \frac{2}{2^{4s}} + \frac{4}{2^{11s} + 2^{9s}} \right) \prod_{p \equiv 1 \pmod{4}} \left(1 + \frac{2}{p^{3s} + p^s} \right) \right. \\ &\quad \left. - \left(1 - \frac{1}{2^{2s}} + \frac{2}{2^{4s}} \right) \right) .\end{aligned}$$

From the above formula, we can easily deduce a crude form of the asymptotic formula:

$$N_4(C_4, X) \sim c(C_4) X^{1/2} \quad \text{with}$$

$$\begin{aligned}c(C_4) &= \frac{3}{\pi^2} \left(\left(1 + \frac{\sqrt{2}}{24} \right) \prod_{p \equiv 1 \pmod{4}} \left(1 + \frac{2}{p^{3/2} + p^{1/2}} \right) - 1 \right) \\ &= 0.12205267325139676092260805289654\dots\end{aligned}$$

It is easy to refine this to the more precise result

$$N_4(C_4, X) = c(C_4) X^{1/2} + O(X^\alpha)$$

for any $\alpha > 1/3$.

In view of the positions of the poles of the function $\Phi_4(C_4, s)$, as for the case $G \simeq S_3$, it should be easy to prove that there is an additional main term, and it is reasonable to conjecture that in fact

$$N_4(C_4, X) = c(C_4) X^{1/2} + c'(C_4) X^{1/3} + O(X^\alpha)$$

for any $\alpha > 1/5$, with

$$\begin{aligned}c'(C_4) &= \frac{3 + 2^{-1/3} + 2^{-2/3}}{1 + 2^{-2/3}} \frac{\zeta(2/3)}{4\pi\zeta(4/3)} \prod_{p \equiv 1 \pmod{4}} \left(1 + \frac{2}{p + p^{1/3}} \right) \left(\frac{1 - 1/p}{1 + 1/p} \right) \\ &= -0.11567519939427878830185483678\dots\end{aligned}$$

5.2 Numerical Computation

Using the Dirichlet series given above for $\Phi_4(C_4, s)$, it is easy to obtain the formula

$$N_4(C_4, X) = (S(X) + S(X/16) + 2S(X/64) + 4S(X/2048) - N_2(C_2, X^{1/2}) - 1)/2 ,$$

where $N_2(C_2, X)$ is given in Section 2 and

$$S(X) = \sum_{\substack{n \leq X^{1/3} \\ p|n \Rightarrow p \equiv 1 \pmod{4}}} |\mu(n)| 2^{\omega(n)} \sum_{\substack{m \leq (X/n^3)^{1/2} \\ \gcd(m, 2n)=1}} |\mu(m)| .$$

This formula can be improved in several technical ways, but basically it is the one that we have used.

5.3 Table

In this table, we let $P_4(C_4, X) = \lfloor c(C_4) X^{1/2} + c'(C_4) X^{1/3} \rfloor$ be the predicted value and $E_4(C_4, X) = (N_4(C_4, X) - P_4(C_4, X))/X^{1/5}$ rounded to 5 decimals.

X	$N_4(C_4, X)$	$P_4(C_4, X)$	$E_4(C_4, X)$
10^1	0	0	0
10^2	0	1	-0.39811
10^3	1	3	-0.50238
10^4	10	10	0
10^5	32	33	-0.10000
10^6	113	110	0.18929
10^7	363	361	0.07962
10^8	1168	1167	0.02512
10^9	3732	3744	-0.19019
10^{10}	11930	11956	-0.26000
10^{11}	38045	38060	-0.09464
10^{12}	120925	120896	0.11545
10^{13}	383500	383472	0.07033
10^{14}	1215198	1215158	0.06340
10^{15}	3848219	3848077	0.14200
10^{16}	12180240	12180346	-0.06688
10^{17}	38542706	38542753	-0.01871
10^{18}	121936924	121936998	-0.07400
10^{19}	385715463	385715227	0.16078

6 Degree 4 Fields with $G \simeq V_4 = C_2 \times C_2$

6.1 Dirichlet Series and Asymptotic Formulas

By studying discriminants of biquadratic quartic extensions, it can easily be shown that

$$\begin{aligned} \Phi_4(V_4, s) &= \frac{1}{6} \left(1 + \frac{3}{2^{4s}} + \frac{6}{2^{6s}} + \frac{6}{2^{8s}} \right) \prod_{p \equiv 1 \pmod{2}} \left(1 + \frac{3}{p^{2s}} \right) \\ &\quad - \frac{1}{2} \left(1 + \frac{1}{2^{4s}} + \frac{2}{2^{6s}} \right) \prod_{p \equiv 1 \pmod{2}} \left(1 + \frac{1}{p^{2s}} \right) + \frac{1}{3}. \end{aligned}$$

From the above formula, we can easily deduce a crude form of the asymptotic formula:

$$N_4(V_4, X) \sim c(V_4) X^{1/2} \log^2 X \quad \text{with}$$

$$c(V_4) = \frac{23}{960} \prod_p \left(\left(1 + \frac{3}{p} \right) \left(1 - \frac{1}{p} \right)^3 \right) = 0.0027524302227554813966383118376\dots$$

It is easy to refine this to the more precise result

$$N_4(V_4, X) = (c(V_4) \log^2 X + c'(V_4) \log X + c''(V_4))X^{1/2} + O(X^\alpha)$$

for any $\alpha > 1/3$, with

$$\begin{aligned} c'(V_4) &= 12c(V_4) \left(\gamma - \frac{1}{3} + \frac{9 \log 2}{23} + 4 \sum_{p \geq 3} \frac{\log p}{(p-1)(p+3)} \right) \\ c''(V_4) &= \frac{c'(V_4)^2}{4c(V_4)} - \frac{3}{\pi^2} \\ &\quad + 24c(V_4) \left(\frac{1}{6} - \gamma_1 - \frac{\gamma^2}{2} - \frac{340}{529} \log^2 2 - 4 \sum_{p \geq 3} \frac{p(p+1) \log^2 p}{(p-1)^2(p+3)^2} \right), \end{aligned}$$

where γ is Euler's constant and

$$\gamma_1 = \lim_{n \rightarrow \infty} \left(\sum_{k=1}^n \frac{\log k}{k} - \frac{\log^2 n}{2n} \right) = -0.0728158454836767248605863758749\dots$$

Numerically, we have

$$\begin{aligned} c'(V_4) &= 0.05137957621042353770883347445\dots \\ c''(V_4) &= -0.2148583422482281175118362061\dots \end{aligned}$$

It is reasonable to conjecture that we can in fact take any $\alpha > 1/4$ in the above asymptotic formula.

6.2 Numerical Computation

From the Dirichlet series given above for $\Phi_4(V_4, s)$, we obtain easily that

$$N_4(V_4, X) = \frac{1}{6} \sum_{\substack{n \leq \sqrt{X} \\ n \text{ odd}}} f_X(n) |\mu(n)| 3^{\omega(n)} - \frac{1}{2} N_2(C_2, \sqrt{X}) - \frac{1}{6},$$

where

$$f_X(n) = \begin{cases} 16 & \text{if } 1 \leq n \leq \sqrt{X}/16 \\ 10 & \text{if } \sqrt{X}/16 < n \leq \sqrt{X}/8 \\ 4 & \text{if } \sqrt{X}/8 < n \leq \sqrt{X}/4 \\ 1 & \text{if } \sqrt{X}/4 < n \leq \sqrt{X}. \end{cases}$$

Together with the formula given for $N_2(C_2, X)$ in Section 2, this is the formula that we have used.

6.3 Table

In this table, we let $P_4(V_4, X) = \lfloor (c(V_4) \log^2 X + c'(V_4) \log X + c''(V_4)) X^{1/2} \rceil$ be the predicted value and $E_4(V_4, X) = (N_4(V_4, X) - P_4(V_4, X))/X^{1/4}$ rounded to 5 decimals.

X	$N_4(V_4, X)$	$P_4(V_4, X)$	$E_4(V_4, X)$
10^1	0	0	0
10^2	0	1	-0.31623
10^3	8	9	-0.17783
10^4	47	49	-0.20000
10^5	243	234	0.50611
10^6	1014	1020	-0.18974
10^7	4207	4201	0.10670
10^8	16679	16655	0.24000
10^9	64316	64255	0.34303
10^{10}	242710	242751	-0.12965
10^{11}	901557	901967	-0.72909
10^{12}	3306085	3306219	-0.13400
10^{13}	11982067	11982984	-0.51567
10^{14}	43017383	43016720	0.20966
10^{15}	153156284	153154732	0.27599
10^{16}	541382988	541386997	-0.40090
10^{17}	1901705324	1901714182	-0.49812
10^{18}	6642813777	6642812780	0.03153
10^{19}	23087994312	23087989990	0.07686

7 Degree 4 Fields with $G \simeq D_4$

7.1 Dirichlet Series and Asymptotic Formulas

In [3] we prove that

$$\Phi_4(D_4, s) = \sum_D \frac{1}{|D|^{2s}} (\Phi_{2,D}(C_2, s) - 1) - \frac{3}{2} \Phi_4(V_4, s) - \frac{1}{2} \Phi_4(C_4, s) ,$$

where we sum over all quadratic discriminants D and the formulas for $\Phi_4(V_4, s)$ and $\Phi_4(C_4, s)$ can be found in Sections 6 and 5 respectively. The Dirichlet series $\Phi_{2,D}(C_2, s)$ is defined as follows. Set $K = \mathbb{Q}(\sqrt{D})$, let $r_2(K)$ be the number of nonreal places of K (i.e., 1 if $D < 0$ and 0 if $D > 0$), and denote by $\zeta_K(s)$ the Dedekind zeta function of K . Then

$$\Phi_{2,D}(C_2, s) = \frac{1}{2^{4s-2+r_2(K)} \zeta_K(2s)} \sum_{\mathfrak{c}|2} \mathcal{N} \mathfrak{c}^{2s} \sum_{\chi} \prod_{\mathfrak{p}} \left(1 - \frac{\chi(\mathfrak{p})}{\mathcal{N} \mathfrak{p}^s} \right)^{-1},$$

where χ runs over all quadratic characters of the ray class group $Cl_{\mathfrak{c}^2}(K)$ corresponding to the modulus \mathfrak{c}^2 .

From the above formula, we can easily deduce a crude form of the asymptotic formula:

$$N_4(D_4, X) \sim c(D_4) X \quad \text{with}$$

$$c(D_4) = \frac{3}{\pi^2} \sum_D \frac{1}{2^{r_2(D)} D^2} \frac{L(1, (\frac{D}{\cdot}))}{L(2, (\frac{D}{\cdot}))} = \frac{3}{\pi^2} \sum_{n \geq 1} \frac{1}{n^2} \sum_{\substack{D|n \\ D \in \mathbb{Z}}} \frac{1}{2^{r_2(D)}} \left(\frac{D}{n/|D|} \right) \phi(n/|D|)$$

where the sums on D are over all discriminants D of quadratic fields, $r_2(D) = r_2(\mathbb{Q}(\sqrt{D}))$ as above, and finally $L(s, (\frac{D}{\cdot}))$ is the L -series associated to the Legendre-Kronecker symbol $(\frac{D}{n})$.

It is possible that the constant $c(D_4)$ can be expressed as a finite linear combination of Euler products with explicit coefficients, but we have not been able to find such an expression. Consequently, we must sum over all quadratic discriminants to obtain a numerical value, and use standard extrapolation techniques. In this way, we obtain numerically

$$c(D_4) = 0.052326011\dots$$

where the last digit may be wrong.

In view of the way in which we have obtained the Dirichlet series $\Phi_4(D_4, s)$, we can conjecture that more precisely

$$\begin{aligned} N_4(D_4, X) &= c(D_4) X - \frac{3}{2} \left(c(V_4) \log^2 X + c'(V_4) \log X + c''(V_4) + \frac{c(C_4)}{3} \right) X^{1/2} \\ &\quad + O(X^\alpha) \end{aligned}$$

for all $\alpha > 1/2$. The numerical data suggest that the error term can be replaced by $(c'(D_4) \log X + c''(D_4))X^{1/2} + o(X^{1/2})$ for suitable constants $c'(D_4)$ and $c''(D_4)$.

7.2 Numerical Computation

Using the methods explained in [6], it is not difficult to count the number of quadratic extensions of a given base field K , since these are of the form $K(\sqrt{\alpha})$ for suitable values of α . Hence $N_4(D_4, X)$ is equal to the sum over all fundamental discriminants $D \neq 1$ of the number of quadratic extensions of $\mathbb{Q}(\sqrt{D})$ whose relative ideal discriminant has a norm less than or equal to X/D^2 .

7.3 Tables

In the first table, we let

$$P_4(D_4, X) = \left[c(D_4) X - \frac{3}{2} \left(c(V_4) \log^2(X) + c'(V_4) \log X + c''(V_4) + \frac{c(C_4)}{3} \right) X^{1/2} \right]$$

be the predicted value and $E_4(D_4, X) = (N_4(D_4, X) - P_4(D_4, X))/X^{1/2}$ rounded to 5 decimals. However, in view of the way in which we have obtained the asymptotic formula, we can also set $N_4(I, X) = N_4(D_4, X) + \frac{3}{2}N_4(V_4, X) + \frac{1}{2}N_4(C_4, X)$ (where I stands for imprimitive), and compare it with $P_4(I, X) = \lfloor c(D_4) X \rfloor$. Hence, in the second table, we set $E_4(I, X) = (N_4(I, X) - P_4(I, X))/X^{1/2}$ rounded to 5 decimals.

X	$N_4(D_4, X)$	$P_4(D_4, X)$	$E_4(D_4, X)$	X	$N_4(I, X)$	$P_4(I, X)$	$E_4(I, X)$
10^1	0	1	-0.31623	10^1	0	1	-0.31623
10^2	0	3	-0.30000	10^2	0	5	-0.50000
10^3	24	38	-0.44272	10^3	36.5	52	-0.49015
10^4	413	443	-0.30000	10^4	488.5	523	-0.34500
10^5	4764	4862	-0.30990	10^5	5144.5	5233	-0.27986
10^6	50496	50734	-0.23800	10^6	52073.5	52326	-0.25250
10^7	516399	516766	-0.11606	10^7	522891	523260	-0.11669
10^8	5205848	5207008	-0.11600	10^8	5231450.5	5232601	-0.11505
10^9	52225424	52227698	-0.07191	10^9	52323764	52326011	-0.07106
10^{10}	522889160	522889735	-0.00721	10^{10}	523259190	523259964	-0.00920

In these tables, the behavior of the error term suggests that there is an additional main term, that our theoretical methods are unable to explain. As mentioned above, it is probably of the form $(c'(D_4) \log X + c''(D_4))X^{1/2}$ for suitable values of $c'(D_4)$ and $c''(D_4)$.

8 Degree 4 Fields with $G \simeq A_4$

8.1 Dirichlet Series and Asymptotic Formulas

Using our usual Kummer-theoretic method, we have obtained an explicit expression for the Dirichlet series $\Phi_4(A_4, s)$ which involves a sum over quadratic characters of ray class groups of cyclic cubic fields with modulus dividing 4, and which is too long to be given here.

From this expression, we can easily deduce a crude form of the asymptotic formula:

$$N_4(A_4, X) \sim c(A_4) X^{1/2} \log X \quad \text{with}$$

$$c(A_4) = \lim_{N \rightarrow \infty} \frac{1}{3 \log 2\zeta(3)} \sum_{\substack{K_3 \\ N < f(K_3) \leq 2N}} \frac{h(K_3)R(K_3)c_2(K_3)c_r(K_3)}{f(K_3)^2} P(K_3)$$

with

$$P(K_3) = \prod_{p \text{ split in } K_3} \frac{(1 + 3/p)(1 - 1/p)^2}{1 + 1/p + 1/p^2},$$

where K_3 ranges over all cyclic cubic extensions of \mathbb{Q} up to isomorphism (which can easily be described explicitly, for example by using the Dirichlet series for

C_3 extensions given in Section 3), $f(K_3)$, $h(K_3)$, $R(K_3)$ denote the conductor, class number and regulator of K_3 ,

$$c_r(K_3) = \prod_{p|f(K_3)} \frac{1}{1 + 1/p + 1/p^2}$$

and $c_2(K_3) = 11/8$ if 2 is inert in K_3 , while $c_2(K_3) = 23/20$ if 2 is totally split in K_3 .

As for the case $G \simeq D_4$, it is possible that the constant $c(A_4)$ can be expressed as a finite linear combination of Euler products with explicit coefficients, but we have not been able to find such an expression. Consequently, we must sum over all cyclic cubic extensions of \mathbb{Q} to obtain a numerical value. We obtain the very poor value

$$c(A_4) = 0.017892$$

It should not be too difficult to obtain an improvement of the asymptotic formula to

$$N_4(A_4, X) = (c(A_4) \log X + c'(A_4)) X^{1/2} + O(X^\alpha)$$

for $\alpha > 1/3$. The tables seem to give something like $c'(A_4) = -0.12354$.

8.2 Numerical Computation

We could have used directly the Dirichlet series mentioned above. However for simplicity and also because we lose at most a time factor of 10, we have preferred to generate A_4 extensions using Kummer theory of quadratic extensions over cyclic cubic fields and keep only those extensions whose discriminant is less than the required bound (see [4] for details).

8.3 Table

In this table, we let $P_4(A_4, X) = \lfloor c(A_4) X^{1/2} \log X \rfloor$ be the predicted value and $E_4(A_4, X) = (N_4(A_4, X) - P_4(A_4, X))/X^{1/3}$ rounded to 5 decimals.

X	$N_4(A_4, X)$	$P_4(A_4, X)$	$E_4(A_4, X)$
10^1	0	0	0
10^2	0	0	0
10^3	0	0	0
10^4	4	4	0
10^5	27	26	0.02154
10^6	121	124	-0.03000
10^7	514	521	-0.03249
10^8	2010	2060	-0.10772
10^9	7699	7818	-0.11900
10^{10}	28759	28844	-0.03945
10^{11}	104766	104241	0.11311

9 Degree 4 Fields with $G \simeq S_4$

9.1 Dirichlet Series and Asymptotic Formulas

By using similar methods to the A_4 case but this time with Kummer theory over noncyclic cubic fields, we have computed explicitly the Dirichlet series $\Phi_4(S_4, s)$, which is quite similar in form to $\Phi_4(A_4, s)$. We can easily deduce from this Dirichlet series the crude asymptotic formula

$$N_4(S_4, X) \sim c(S_4) X$$

but the expression for the constant $c(S_4)$ is too complicated to be given here and is not easily computed numerically since we must sum over all noncyclic cubic extensions of \mathbb{Q} . Hence, contrary to the other Galois groups, it is for the moment difficult to give similar tables to the ones given above.

As for the case $G \simeq D_4$, it is possible that the constant $c(S_4)$ can be expressed as a finite linear combination of Euler products with explicit coefficients, but we have not been able to find such an expression. Wright and Yukie (private communication) assert that they have such an expression, not yet completely proved, but the error term is so large that the amount of data that we have is insufficient to check whether their expression is plausible. A combination of their work with experimental data suggests that we could have (but this is to be taken with a huge grain of salt)

$$N_4(S_4, X) = 0.6382 X - 0.764 X^{0.97} + O(X^\alpha)$$

for some $\alpha < 0.97$, perhaps any $\alpha > 1/2$.

9.2 Numerical Computation

As for A_4 extensions, we use Kummer theory of quadratic extensions, this time over noncyclic cubic fields and we keep only those extensions whose discriminant is less than the required bound. See [4] for details.

9.3 Table

As mentioned above, we give the exact values of $N_4(S_4, X)$ together with the value $P_4(S_4, X) = [0.6382 X - 0.764 X^{0.97}]$ and $E_4(S_4, X) = (N_4(S_4, X) - P_4(S_4, X))/X^{1/2}$, but it should be once again emphasized that contrary to the other Galois groups, these cannot be considered as predictions but just as guesses.

X	$N_4(S_4, X)$	$P_4(S_4, X)$	$E_4(S_4, X)$
10^1	0	-1	0.31623
10^2	0	-3	0.30000
10^3	18	17	0.03162
10^4	570	586	-0.16000
10^5	9739	9733	0.01897
10^6	133322	133430	-0.10800

References

1. K. Belabas, On the mean 3-rank of quadratic fields, *Compositio Math.* **118** (1999), 1–9.
2. K. Belabas, A fast algorithm to compute cubic fields, *Math. Comp.* **66** (1997), 1213–1237.
3. H. Cohen, F. Diaz y Diaz and M. Olivier, Density of number field discriminants, in preparation.
4. H. Cohen, F. Diaz y Diaz and M. Olivier, Construction of tables of quartic fields using Kummer theory, this volume.
5. H. Cohen, A course in computational algebraic number theory (third printing), GTM **138**, Springer-Verlag, 1996.
6. H. Cohen, Advanced topics in computational number theory, GTM **193**, Springer-Verlag, 2000.
7. B. Datskovsky and D. J. Wright, Density of discriminants of cubic extensions, *J. reine angew. Math.* **386** (1988), 116–138.
8. H. Davenport and H. Heilbronn, On the density of discriminants of cubic fields I, *Bull. London Math. Soc.* **1** (1969), 345–348.
9. H. Davenport and H. Heilbronn, On the density of discriminants of cubic fields II, *Proc. Royal Soc. A* **322** (1971), 405–420.
10. D. Roberts, Density of cubic field discriminants, Algebraic number theory preprint archive **177** (April 26, 1999).
11. T. Shintani, On Dirichlet series whose coefficients are class numbers of integral binary cubic forms, *J. Math. Soc. Japan* **24** (1972), 132–188.
12. T. Shintani, On zeta-functions associated with the vector space of quadratic forms, *J. Fac. Sci. Univ. Tokyo, Sec. 1a* **22** (1975), 25–66.
13. G. Tenenbaum, Introduction à la théorie analytique et probabiliste des nombres, Cours Spécialisés SMF **1**, Société Mathématique de France, 1995.
14. D. J. Wright, Distribution of discriminants of Abelian extensions, *Proc. London Math. Soc.* (3) **58** (1989), 17–50.
15. D. J. Wright and A. Yukie, Prehomogeneous vector spaces and field extensions, *Invent. Math.* **110** (1992), 283–314.

On Reconstruction of Algebraic Numbers

Claus Fieker and Carsten Friedrichs

Technische Universität Berlin
Fachbereich 3, Sekr. MA 8-1
Straße des 17. Juni 136, 10623 Berlin, Germany
`{fieker,fried}@math.TU-Berlin.de`

Abstract. Let L be a number field and \mathfrak{a} be an ideal of some order of L . Given an algebraic number $a \pmod{\mathfrak{a}}$ and some bounds we show how to effectively reconstruct a number b if it exists such that b is smaller than the given bound and $b \equiv a \pmod{\mathfrak{a}}$.

The first application is an algorithm for the computation of n -th roots of algebraic numbers. Secondly, we get an algorithm to factor polynomials over number fields which generalizes the Hensel-factoring method. Our method uses only integral LLL-reductions in contrast to the real LLL-reductions suggested by [6,8].

1 Introduction

One of the most basic methods in algorithmic number theory is to work “modular”, i.e. instead of looking for a solution $\alpha \in \mathbb{Z}_F$ the ring of integers (or any other order) of the number field F , one examines the problem modulo a suitable ideal \mathfrak{a} . More precisely, one looks at the canonical epimorphism $\phi : R \rightarrow R/\mathfrak{a}$ and considers the “easier” problem in R/\mathfrak{a} .

Suppose we have a solution $\beta \pmod{\mathfrak{a}}$ and want to lift it to a solution α (“reconstruct” α from β). In order to do this we need some additional information. In this paper we focus on lattice based techniques, the additional information will be a bound on the “size” of α . This enables us to find α as a smallest element in the coset $\beta + \mathfrak{a} -$ provided \mathfrak{a} is “large” enough.

This idea has already been used in the literature ([3,6,8]). The difference to our approach however is the use of a different lattice (a different size function). Our choice of the lattice allows us to work with integers only.

This new reconstruction has already been used successfully for factoring polynomials over number fields [6] and computation of roots in class field computations. Other applications are the computation of embeddings of number fields and irreducibility testing for polynomials.

We don’t use the canonical Minkowski-map to embed the ring of integers into a real vector space. Our embedding depends on the chosen \mathbb{Z} -basis of the ring of integers. Although this seems to be a theoretical disadvantage it yields a good computational behavior. We have implemented our method in the computer algebra system **KASH** [4] and provide some numerical examples which point out the advantages of our method at the end of this article.

2 Successive Minima

Let us fix some notation first. Through the remainder of this article K will be a number field of degree n (over \mathbb{Q}) and R will be a fixed order of K . All lattices (which will be denoted by Δ in the following) are considered to be subsets of \mathbb{R}^n and are equipped with the Euclidean norm $\|\cdot\|$ as length.

Further we choose a \mathbb{Z} -isomorphism (group-isomorphism)

$$\delta : R \rightarrow \Delta . \quad (1)$$

By Q (or Q_Δ if we want to emphasize the lattice involved) we denote the quadratic form, occasionally viewed as a mapping $Q_\Delta : R \rightarrow \mathbb{R}^{\geq 0}$, which is achieved by applying the scalar product (of \mathbb{R}^n) to the image of the \mathbb{Z} -isomorphism δ . The first successive minimum $\lambda_1(\Delta)$ is the square of the length of the shortest non-zero element in the lattice Δ .

We fix a basis $\omega_1, \dots, \omega_n$ for R as a \mathbb{Z} -module, this also yields a \mathbb{Q} -basis for the number field K . For an integral ideal \mathfrak{a} of R , $\Delta(\mathfrak{a})$ denotes the sublattice of Δ corresponding to \mathfrak{a} as a submodule of R .

Let us fix an element $\beta \in R$ and some real number $c > 0$. Our task is now to decide if there exists any $\alpha \in R$ such that $\alpha - \beta \in \mathfrak{a}$ and $Q(\alpha) \leq c$.

We begin with the following trivial lemma:

Lemma 1. *Suppose the first successive minimum $\lambda_1(\Delta(\mathfrak{a}))$ is greater than $4c$. Then there exists at most one $\alpha \in R$ such that $Q(\alpha) \leq c$ and $\beta - \alpha \in \mathfrak{a}$.*

Proof. Suppose we have α_1 and α_2 with the desired properties. Then $\alpha_1 - \alpha_2 = \alpha_1 - \beta + \beta - \alpha_2 \in \mathfrak{a}$ and $\sqrt{Q(\alpha_1 - \alpha_2)} \leq \sqrt{Q(\alpha_1)} + \sqrt{Q(\alpha_2)} < 2\sqrt{c}$, implying $Q(\alpha_1 - \alpha_2) < 4c \leq \lambda_1(\Delta(\mathfrak{a}))$. Therefore we get $0 = \alpha_1 - \alpha_2$. \square

Remark 1. Under the assumptions of lemma 1, it is possible to compute α as the lattice point closest to β by well known enumeration procedures [7].

Since the enumeration part has potentially exponential running time and is therefore potentially slow, we show how to avoid enumeration. We need another lemma from [3]:

Lemma 2. *Suppose b_1, \dots, b_n is a LLL-reduced basis of the lattice Δ . Then we obtain $\min_{\mu \in \mathbb{R}^n, \|\mu\|=1} Q(\sum \mu_i \delta^{-1}(b_i)) \geq \lambda_1(\Delta) 2^{-n(n-1)/4}$.*

The LLL-reduction [5] can be found in most text books on computational algebraic number theory, e.g. [7] which also contains an algorithm for computing a LLL-reduced lattice basis. The idea of the LLL-reduction is to construct a basis which is as orthogonal as possible. Let us remark that a basis of the order R or the integral ideal \mathfrak{a} induces a basis of the corresponding lattice and vice versa. The matrix transforming the lattice basis to a LLL-reduced lattice basis can also be applied to get the corresponding order basis resp. ideal basis.

Having the above lemma in mind, the following is straightforward:

Lemma 3. Suppose $c < \lambda_1(\Delta(\mathfrak{a}))2^{-n(n-1)/4-2}$, b_1, \dots, b_n is a LLL-basis for $\Delta(\mathfrak{a})$ and let $\beta = \sum_{i=1}^n r_i \delta^{-1}(b_i)$ with $r_i \in \mathbb{Q}$. If there exists $\alpha \in \beta + \mathfrak{a}$ with $Q(\alpha) < c$ then $\alpha = \sum_{i=1}^n (r_i - \lfloor r_i \rfloor) \delta^{-1}(b_i)$.

(We write $\lfloor r \rfloor = z$ to denote the unique $z \in [r - 1/2, r + 1/2] \cap \mathbb{Z}$.)

3 Reconstruction Using the Method of Pohst–Roblot

In order to use the ideas of the preceding section we have to specify the lattice and to show how to choose the ideal which amounts to give an estimate for $\lambda_1(\mathfrak{a})$. Both Pohst [6] and Roblot [8] use the Minkowski-map to map R into \mathbb{R}^n : Let $F = \mathbb{Q}(\gamma)$ be given via a primitive element γ . The conjugates $\gamma^{(i)}$ of γ are ordered in the usual way: $\gamma^{(1)}, \dots, \gamma^{(r_1)} \in \mathbb{R}$ and $\gamma^{(r_1+1)} = \overline{\gamma^{(r_1+r_2+1)}}, \dots, \gamma^{(r_1+r_2)} = \overline{\gamma^{(r_1+r_2+r_2)}} \in \mathbb{C} \setminus \mathbb{R}$. Then we define

$$\delta_{\mathbb{R}}(\gamma)_{(i)} := \begin{cases} \gamma^{(i)} & 1 \leq i \leq r_1 \\ \sqrt{2}\Re(\gamma^{(i)}) & r_1 + 1 \leq i \leq r_1 + r_2 \\ \sqrt{2}\Im(\gamma^{(i-r_2)}) & r_1 + r_2 + 1 \leq i \leq n \end{cases} \quad (2)$$

and $\Delta_{\mathbb{R}} := \delta_{\mathbb{R}}(R)$. We get $Q_{\Delta_{\mathbb{R}}}(\alpha) = Q_{\mathbb{R}}(\alpha) := T_2(\alpha) := \sum_{i=1}^n |\alpha^{(i)}|^2$.

For the reconstruction process Pohst applies lemma 1 whereas Roblot makes use of lemma 3. In addition, both use the following:

Lemma 4. For every non zero $\gamma \in \mathfrak{a}$ we have $T_2(\gamma) \geq nN(\mathfrak{a})^{2/n}$.

Proof. We have $N(\gamma) \geq N(\mathfrak{a})$. Now, by the inequality between geometric and arithmetic means we get:

$$(N(\gamma))^{2/n} = \left(\prod_{i=1}^n \gamma^{(i)} \right)^{2/n} \leq \frac{1}{n} T_2(\gamma) . \quad (3)$$

□

Using ideals of the form \mathfrak{p}^k for a fixed prime ideal \mathfrak{p} it is now easy to compute the exponent necessary to ensure $\lambda_1(\mathfrak{p}^k) > d$ for any $d > 0$.

As demonstrated in [6,8] this can be used to get an efficient factoring algorithm for polynomials over number fields.

Since both Pohst [6] and Roblot [8] work with real lattices, their algorithms suffer from the usual problems with real arithmetic in computer algebra systems.

4 The New Approach

To overcome the precision problems, we choose a different lattice. Since R is a free \mathbb{Z} -module of rank n , it is natural to consider $\Delta_{\mathbb{Z}} := \mathbb{Z}^n$ with the usual scalar product. $Q_{\mathbb{Z}}(\alpha)$ now measures the size of the coefficients of α if represented as a linear combination of $\omega_1, \dots, \omega_n$.

The two isomorphisms $\delta_{\mathbb{Z}} : R \rightarrow \Delta_{\mathbb{Z}}$ and $\delta_{\mathbb{R}} : R \rightarrow \mathbb{R}^n$ induce a third isomorphism $\psi : \Delta_{\mathbb{Z}} \rightarrow \Delta_{\mathbb{R}} : \psi := \delta_{\mathbb{R}} \circ \delta_{\mathbb{Z}}^{-1}$. This isomorphism is also obtained using the “real” basis of R : $\psi(x) = M \cdot x$, where $M = (\delta_{\mathbb{R}}(\omega_1), \dots, \delta_{\mathbb{R}}(\omega_n))$ for a fixed basis $\omega_1, \dots, \omega_n$ of R .

The quadratic form on $\Delta_{\mathbb{R}}$ resp. $\Delta_{\mathbb{Z}}$ will be denoted by $Q_{\mathbb{R}}(\cdot)$ resp. $Q_{\mathbb{Z}}(\cdot)$.

Since ψ and ψ^{-1} are continuous (when considered as maps from $\mathbb{R}^n \rightarrow \mathbb{R}^n$) we get constants $c_1, c_2 \in \mathbb{R}$ such that $Q_{\mathbb{R}}(x) = \|\delta_{\mathbb{R}}(x)\|^2 = \|\psi \circ \delta_{\mathbb{Z}}(x)\|^2 \leq c_1 \|\delta_{\mathbb{Z}}(x)\|^2 = c_1 Q_{\mathbb{Z}}(x)$ for all $x \in R$ and $Q_{\mathbb{Z}}(y) = \|\delta_{\mathbb{Z}}(y)\|^2 = \|\psi^{-1} \circ \delta_{\mathbb{R}}(y)\|^2 \leq c_2 \|\delta_{\mathbb{R}}(y)\|^2 = c_2 Q_{\mathbb{R}}(y)$ for all $y \in R$. It is well known from (numerical) analysis, (see e.g. [9]) that the smallest possible c_i are obtained as the largest (c_1) resp. smallest (c_2^{-1}) eigenvalue of $M^t \cdot M$. We note that the eigenvalues of $M^t \cdot M$ are real and positive.

Now we have to consider the sublattice $\Delta_{\mathbb{Z}}(\mathfrak{a})$ corresponding to the ideal \mathfrak{a} as a submodule of the ring R . The sublattice $\Delta_{\mathbb{Z}}(\mathfrak{a})$ is generated by the columns of the “transformation” matrix $B \in \mathbb{Z}^{n \times n}$ i.e. a basis of the \mathbb{Z} -module \mathfrak{a} is obtained via $(a_1, \dots, a_n) = (\omega_1, \dots, \omega_n) \cdot B^t$, and the columns $B_{1 \cdot}, \dots, B_{n \cdot}$ of B form a basis of the sublattice $\Delta_{\mathbb{Z}}(\mathfrak{a})$.

Provided with this we are able to apply lemma 1 and 3, i.e. given an upper bound for the size of the coefficients of α and an ideal with first successive minimum satisfying the condition of lemma 1, then we can reconstruct α via enumeration in the lattice $\Delta_{\mathbb{Z}}(\mathfrak{a})$. If the condition of lemma 3 is satisfied, we can reconstruct α in the LLL-reduced basis of $\Delta_{\mathbb{Z}}(\mathfrak{a})$ via rounding of the coefficients ($\lfloor \cdot \rfloor$).

So we need a way to estimate the first successive minimum of $\Delta_{\mathbb{Z}}(\mathfrak{a})$.

Lemma 5. *The following lower bound for the first successive minimum holds:*

$$\lambda_1(\Delta_{\mathbb{Z}}(\mathfrak{a})) \geq \frac{n}{c_1} N(\mathfrak{a})^{2/n} \quad (4)$$

Proof.

$$\begin{aligned} \lambda_1(\Delta_{\mathbb{Z}}(\mathfrak{a})) &= \min_{z \in \mathfrak{a} \setminus \{0\}} Q_{\mathbb{Z}}(z) \geq \min_{z \in \mathfrak{a} \setminus \{0\}} \frac{1}{c_1} Q_{\mathbb{R}}(z) \\ &= \frac{1}{c_1} \lambda_1(\Delta_{\mathbb{R}}(\mathfrak{a})) \geq \frac{1}{c_1} n N(\mathfrak{a})^{2/n}. \end{aligned}$$

The last statement is a consequence of lemma 4. □

If we are given an upper bound for the T_2 -value of α , we can easily calculate an upper bound for the size of the coefficients of α .

Lemma 6. *Provided that $T_2(\gamma) \leq c$ for $\gamma \in R$ we have $Q_{\mathbb{Z}}(\gamma) \leq c_2 \cdot c$.*

Proof. Immediate. □

Collecting all this we get the following theorem:

Theorem 1. Let \mathfrak{a} be an ideal such that

$$N(\mathfrak{a}) > \left(\frac{c_1 c_2 c 2^{n(n-1)/4+2}}{n} \right)^{n/2} \quad (5)$$

and b_1, \dots, b_n be a LLL-reduced basis of the lattice $\Delta_{\mathbb{Z}}(\mathfrak{a})$. Let $\beta \in R$ be arbitrary. Then $\beta = \sum_{i=1}^n q_i \delta_{\mathbb{Z}}^{-1}(b_i)$ with some $q_i \in \mathbb{Q}$. Furthermore, we assume that there is an $\alpha \in \beta + \mathfrak{a}$ such that $Q_{\mathbb{R}}(\alpha) < c$.

Then we get $\alpha = \beta - \gamma$, where $\gamma := \sum_{i=1}^n \lfloor q_i \rfloor \delta_{\mathbb{Z}}^{-1}(b_i)$.

Proof. Assuming that

$$N(\mathfrak{a}) > \left(\frac{c_1 c_2 c 2^{n(n-1)/4+2}}{n} \right)^{n/2} \quad (6)$$

lemma 5 yields

$$\lambda_1(\Delta_{\mathbb{Z}}(\mathfrak{a})) \geq \frac{c_2 c 2^{n(n-1)/4+2}}{n} . \quad (7)$$

Now we apply lemma 3 to obtain the unique element $\gamma := \sum_{i=1}^n \lfloor q_i \rfloor \delta_{\mathbb{Z}}^{-1}(b_i)$ if it exists and conclude that $Q_{\mathbb{Z}}(\beta - \gamma) < cc_2$.

Lemma 6 shows that $\alpha = \beta - \gamma$. \square

This new approach has the advantage of purely integral computations while dealing with the lattice $\Delta_{\mathbb{Z}}$, i.e. LLL-reduction and enumeration of the closest lattice points. For the sake of this we get worse bounds (larger exponents of the prime ideal \mathfrak{p}) than Pohst [6] and Roblot [8]. The computation of the ideal $\mathfrak{a} = \mathfrak{p}^k$ can be done very efficiently using the ideas of [6], so that the worse bounds do not yield a disadvantage of our method. This is also demonstrated by the numerical examples.

5 Denominators

In order to extend our method to the reconstruction of non-integral numbers we have at least two choices: to convert the problem into an integral one (by multiplying everything with a suitable integer, choosing a different polynomial, etc.) and using a bound on the denominator, to reconstruct it. This method is therefore useful for reconstructing elements not contained in the equation order.

We consider the following situation: We want to find α/d with $\alpha \in R$ and $d \in \mathbb{N}$. We know bounds for α and d : $Q_{\mathbb{Z}}(\alpha) < B$ and $d^2 < B$, $d \notin \mathfrak{a}$ and assume that we have already computed $\beta \in R$ such that $d\beta - \alpha \in \mathfrak{a}$.

In order to compute α and d we make again use of the LLL-reduction, we extend the lattice $\Delta_{\mathbb{Z}}$ using the following map:

$$\iota : \mathbb{Z}^n \rightarrow \mathbb{Z}^{n+1} : (z_1, \dots, z_n)^t \mapsto (0, z_1, \dots, z_n)^t . \quad (8)$$

Let $\Delta'(\mathfrak{a}) := \langle \iota(\Delta_{\mathbb{Z}}(\mathfrak{a}), \beta') \rangle$, by β' we denote the vector $(1, \beta_1, \dots, \beta_n)^t$ where $\beta = \sum_{i=1}^n \beta_i \omega_i$ and $\omega_1, \dots, \omega_n$ is the \mathbb{Z} -basis of R . In order to simplify the notation we use Q' instead of $Q_{\Delta'(\mathfrak{a})}$ as quadratic form on the lattice $\Delta'(\mathfrak{a})$.

Lemma 7. *If the ideal \mathfrak{a} has first successive minimum $\lambda_1(\Delta_Z(\mathfrak{a})) > 4B_1^2$, then there is at most one $\omega \in \Delta'(\mathfrak{a})$ subject to $Q'(\omega) < B_1$.*

Proof. Suppose $\omega_1, \omega_2 \in \Delta'(\mathfrak{a})$ with $Q'(\omega_i) < B_1$. The elements ω_i are of the form $\omega_i = f_i\beta' + \iota(\delta_{\mathbb{Z}}(a_i))$ with $f_i \in \mathbb{Z}, a_i \in \mathfrak{a}, i = 1, 2$. An immediate consequence of the bounds $Q'(\omega_i) < B_1$ is $f_i < \sqrt{B_1}$.

There exists an element $\bar{\alpha} \in \Delta_Z(\mathfrak{a})$ such that $\iota(\bar{\alpha}) = f_2\omega_1 + f_1\omega_2 \in \iota(\Delta_Z(\mathfrak{a})) \subseteq \Delta'(\mathfrak{a})$. $\sqrt{Q_{\mathbb{Z}}(\bar{\alpha})} = \sqrt{Q'(\iota(\bar{\alpha}))} = \sqrt{Q'(f_2\omega_1 + f_1\omega_2)} \leq f_2\sqrt{Q'(\omega_1)} + f_1\sqrt{Q'(\omega_2)} < (f_1 + f_2)\sqrt{B_1} < 2B_1$. We conclude $Q_{\mathbb{Z}}(\bar{\alpha}) < 4B_1^2 < \lambda_1(\Delta_Z(\mathfrak{a}))$ such that $\bar{\alpha} = 0$ and $\omega_1 = \omega_2$. \square

Theorem 2. *Suppose that the ideal \mathfrak{a} has first successive minima $\lambda_1(\Delta_Z(\mathfrak{a})) > 16B^2$, and there exists α and d such that $d\beta - \alpha \in \mathfrak{a}$, then the shortest vector of the lattice $\Delta'(\mathfrak{a})$ is $\omega = d\beta' + \iota(\delta_{\mathbb{Z}}(\alpha))$.*

Proof. We know that $\alpha \equiv d\beta \pmod{\mathfrak{a}}$ so there exists an element $\bar{a} \in \mathfrak{a}$ with the property $\alpha = d\beta + \bar{a}$. The element $\omega := d\beta' + \iota(\delta_{\mathbb{Z}}(\bar{a})) = de_1 + \iota(\delta_{\mathbb{Z}}(\alpha))$ where e_1 is the first canonical basis element of \mathbb{Z}^{n+1} . It is easy to see that $Q'(\omega) = d^2 + Q_{\mathbb{Z}}(\alpha) < 2B$. Using $B_1 = 2B$ in lemma 7 we obtain that ω must be the shortest vector in $\Delta'(\mathfrak{a})$. \square

Remark 2. If the bounds of theorem 2 are suitably enlarged, we can use the first basis vector of a LLL-reduced basis for $\Delta'(\mathfrak{a})$ to get a solution.

In contrast to the preceding version (theorem 1) we need a LLL-reduction for each number to be reconstructed. In theorem 1 we only need one LLL-reduction for the ideal basis and can use rounding to reconstruct any number of integers. We want to remark that this method (theorem 2) also uses purely integral computations only, i.e. the lattice $\Lambda' \subset \mathbb{Z}^{n+1}$, because all the $\beta_i \in \mathbb{Z}$.

6 Numerical Improvements

The value $\sqrt{c_1 c_2}$ occurring in theorem 1 is known in numerical analysis as the condition $\text{cond}(M)$ of M . This is used as a measure for the numerical difficulty of the matrix [9].

Since $\text{cond}(M)^n$ is a factor of our bounds, we should try to reduce it. Obviously, $\text{cond}(M) = 1$ if M is an orthogonal matrix. Therefore one should start with a LLL-reduced basis for $\Delta_{\mathbb{R}}(R)$ in order to “make M as orthogonal as possible”. Since $\text{cond}(M)$ is independent of the application (i.e. part of every estimate used) we are permitted to spend some time on the computation (of a good estimate) of $\text{cond}(M)$.

We want to point out three different methods to compute (estimates for) the condition $\text{cond}(M)$, the first is related to algebraic number theory, the second and the third are suggested by numerical analysis [9,10].

6.1 Characteristic Polynomial

Every computer algebra system contains a function to compute the characteristic polynomial and another function which computes the real zeroes of a polynomial. So we can easily make use of this:

1. Compute the characteristic polynomial $\chi_{M^t M}(x)$ of the matrix $M^t M$,
2. compute the real zeros of $\chi_{M^t M}(x)$: $\lambda_n \geq \dots \geq \lambda_1$,
3. the condition will be $\text{cond}(M) = \sqrt{\lambda_n / \lambda_1}$.

6.2 Von-Mises-Iteration

If the matrix $M^t M$ is equivalent to a diagonal matrix and the largest eigenvalue λ_{\max} occurs with frequency one, we can construct a sequence $a_i := \frac{\|t_i\|}{\|t_{i-1}\|}$, where $t_i := A^i t_0$ converging to the largest eigenvalue, provided that the input vector t_0 is not orthogonal to the eigenspace of λ_{\max} . The von-Mises-Iteration is also known as the “power-method” [10, p.47].

We have to apply the von-Mises-Iteration also to the inverse of the matrix $M^t M$, to obtain the smallest eigenvalue λ_{\min} of $M^t M$. This method suffers from the slow convergence in the case λ_{n-1} (the second largest eigenvalue) is very close to λ_{\max} .

6.3 Matrix-Norm

This method is very simple and known as the theorem of Hirsch [10, p.81]:

Theorem 3. *Assuming that the matrix norm $\|\cdot\|$ is admissible for the (Euclidean) vector norm (i.e. $\|Mx\|_2 \leq \|M\| \|x\|_2$), then the inequality $\lambda_{\max} \leq \|M^t M\|$ holds.*

So we can use for example the Schur-norm

$$\|A\|_{\text{Schur}} := \left(\sum_{i,k=1}^n |a_{ik}|^2 \right)^{1/2} \quad (9)$$

which is admissible for the Euclidean vector norm [9, p.167]. This yields the following upper bound for the condition of M :

$$\text{cond}(M) \leq \|M^t M\|_{\text{Schur}} \|(M^t M)^{-1}\|_{\text{Schur}} . \quad (10)$$

7 Applications

In this section we investigate different applications of the new reconstruction method, i.e. factorization of polynomials over number fields, irreducibility-testing of polynomials over number fields and computation of r -th roots of algebraic elements. We demonstrate the algorithms and point out some improvements.

In order to use the developed theory, we apply an algorithm of the following shape:

1. task: compute $\alpha \in R$ with a certain property
2. compute a bound for the result, usually $T_2(\alpha) \leq c$
3. pick a suitable prime ideal \mathfrak{p} of R
4. compute k such that $\mathfrak{a} := \mathfrak{p}^k$ matches either theorem 1 or theorem 2.
5. compute an approximation β of $\alpha \pmod{\mathfrak{a}}$, this usually involves a Hensel- or Newton-lifting
6. use theorem 1 or 2 to either find α or show that it does not exist

Depending on the actual problem the details vary. Suppose we know that α exists. Then we can simply try to compute $\alpha \pmod{\mathfrak{p}^k}$ for increasing k until we are successful. Since the bounds are usually much too large, this gives quite a speedup.

One often tries different prime ideals, e.g. to get a first degree prime ideal (to simplify the lifting) or to get special factoring shapes. We will discuss this for each problem individually.

7.1 Factorization and Irreducibility-Testing

Let $f \in R[x]$ be a square-free monic polynomial and R be the maximal order of some number field F . We want a complete factorization of f . Since f is monic, the factorizations over R and over F coincide. From [6] or [1] we get estimates for the coefficients of a factor g of f as follows:

Theorem 4. Let $f = x^d + \sum_{i=0}^{d-1} a_i x^i \in R[x]$ be a polynomial and $g = x^k + \sum_{i=0}^{k-1} b_i x^i$ a proper factor. Then

$$|b_i^{(s)}| \leq \frac{3^{3/4}}{\sqrt{\pi}} \frac{3^{d/2}}{\sqrt{d}} [f^{(s)}]_2 \quad (11)$$

for all i where $[f^{(s)}]_2 := \sum_{j=0}^d \binom{d}{j}^{-1} |a_j^{(s)}|^2$.

Furthermore, it is well known that if \mathfrak{p} is no divisor of the discriminant of f then $f \pmod{\mathfrak{p}}$ is square free, too. Therefore we can obtain a factorization of f modulo \mathfrak{p}^k using linear or quadratic Hensel-lifting.

The algorithm is straightforward: Compute all possible factors $\pmod{\mathfrak{p}^k}$ for a suitable k , apply reconstruction to each coefficient and test if one gets true factor in $R[x]$ this way. To make this efficient, we use ideas from [2] for early factor detection and some heuristics for the enumeration of all factors.

The Irreducibility-Testing is just a special case of the general factoring algorithm. But the computation can be speed up, if one stops after the first true factor is found.

7.2 Roots

This too can be viewed as a factorization problem. However, due to special shape of the polynomial we get sharp bounds for the factors. Furthermore, since we are only looking for linear factors, we need no recombination step. The lifting necessary for the root computation can be done using quadratic Newton lifting. When working without bounds it is advisable to use $\mathfrak{p}^{(2^k)}$ rather than \mathfrak{p}^k .

8 Examples and Discussion

The numerical examples in this section demonstrate the advantages of the new method. For computations we use the computer algebra system KASH [4] on a Intel Pentium III (600MHz, 512MB RAM) running under Linux 2.2.13-SMP.

In the case of totally real fields one can get rid of the real arithmetic by using LLL-reduction on the gram matrix, which has integral entries in this case, instead of using LLL-reduction on the real-basis of the lattice. The implementation of Pohst's algorithm [6] benefits from this fact. This will be demonstrated by the following two examples which are taken from [6].

8.1 Factorization 1

Let $F := \mathbb{Q}(\gamma)$ generated by a root γ of the polynomial $f = t^{12} + t^{11} - 28t^{10} - 40t^9 + 180t^8 + 426t^7 + 89t^6 - 444t^5 - 390t^4 - 75t^3 + 27t^2 + 11t + 1$. This polynomial splits completely over the number field F .

It takes 170 ms to factor f over F using the algorithm of Pohst [6]. The creation of the real lattice (including the LLL-reduction) takes 50 ms, whereas our new method takes 210 ms.

For the computation both algorithms use the following prime ideal and bounds: $\mathfrak{p} = 65449R + (2545 + \gamma)R$, where R is the ring of integers in F , and a T_2 -bound of 1344 for the coefficients of linear factors yielding an exponent of 4 for the prime ideal \mathfrak{p} .

Using a LLL-reduced basis for R and the complex zeroes for the characteristic polynomial, we obtain $\text{cond}(M) < 4$. Applying the Schur-norm, we get $\text{cond}(M) < 27$. This increases the bound for the exponent to 12 resp. 13.

8.2 Irreducibility-Testing

The polynomial $g := t^{14} + \frac{1}{196}(-2139 + 2510\gamma + 1573\gamma^2 - 1419\gamma^3 + 388\gamma^4 + 760\gamma^5 - 334\gamma^6 - 2043\gamma^7 - 1292\gamma^8 + 495\gamma^9 + 17\gamma^{10} - 12\gamma^{11})t^{13} + \frac{1}{98}(222 - 1517\gamma - 1800\gamma^2 + 1436\gamma^3 + 936\gamma^4 - 2805\gamma^5 + 29\gamma^6 + 375\gamma^7 + 1006\gamma^8 + 97\gamma^9 - 65\gamma^{10} + \gamma^{11})t^{12} + \frac{1}{196}(4798 - 45\gamma - 9\gamma^2 + 702\gamma^3 - 249\gamma^4 - 3125\gamma^5 - 145\gamma^6 - 13\gamma^7 + 2572\gamma^8 + 754\gamma^9 + 31\gamma^{10} + 16\gamma^{11})t^{11} + \frac{1}{196}(3842 + 3333\gamma + 2614\gamma^2 + 1957\gamma^3 - 1135\gamma^4 + 106\gamma^5 + 1614\gamma^6 + 338\gamma^7 + 2295\gamma^8 + 542\gamma^9 - 71\gamma^{10} + 11\gamma^{11})t^{10} + \frac{1}{196}(-4552 - 6396\gamma - 3795\gamma^2 - 1273\gamma^3 - 6386\gamma^4 - 2049\gamma^5 - 377\gamma^6 - 1249\gamma^7 - 2347\gamma^8 - 1310\gamma^9 - 86\gamma^{10} - 13\gamma^{11})t^9 + \frac{1}{196}(3996 + 3774\gamma + 3335\gamma^2 + 4421\gamma^3 + 2750\gamma^4 + 2325\gamma^5 + 1425\gamma^6 - 1643\gamma^7 + 853\gamma^8 + 570\gamma^9 + 62\gamma^{10} - 3\gamma^{11})t^8 + \frac{1}{196}(2266 + 4447\gamma - 904\gamma^2 + 4607\gamma^3 - 919\gamma^4 + 3152\gamma^5 + 1072\gamma^6 - 304\gamma^7 + 1251\gamma^8 + 678\gamma^9 + 5\gamma^{10} + \gamma^{11})t^7 + \frac{1}{196}(1324 + 2288\gamma + 3063\gamma^2 - 1509\gamma^3 - 552\gamma^4 - 4369\gamma^5 + 199\gamma^6 + 495\gamma^7 + 785\gamma^8 + 816\gamma^9 + 22\gamma^{10} + 17\gamma^{11})t^6 + \frac{1}{98}(3347 + 755\gamma + 424\gamma^2 + 1032\gamma^3 + 3431\gamma^4 + 419\gamma^5 + 1025\gamma^6 + 1300\gamma^7 + 64\gamma^8 + 725\gamma^9 - 13\gamma^{10} + 10\gamma^{11})t^5 + \frac{1}{28}(35 - 18\gamma - 662\gamma^2 - 682\gamma^3 - 802\gamma^4 - 229\gamma^5 - 223\gamma^6 - 290\gamma^7 + 177\gamma^8 - 7\gamma^9 - 23\gamma^{10} - \gamma^{11})t^4 + \frac{1}{196}(326 - 6073\gamma + 2018\gamma^2 - 8003\gamma^3 - 5421\gamma^4 - 2972\gamma^5 - 1556\gamma^6 - 568\gamma^7 - 381\gamma^8 - 1328\gamma^9 - 157\gamma^{10} - 9\gamma^{11})t^3 + \frac{1}{196}(730 - 5371\gamma + 4670\gamma^2 - 239\gamma^3 - 2245\gamma^4 + 656\gamma^5 + 440\gamma^6 - 508\gamma^7 - 1251\gamma^8 - 552\gamma^9 - 89\gamma^{10} - \gamma^{11})t^2 + \frac{1}{196}(2930 + 3111\gamma + 4720\gamma^2 + 859\gamma^3 - 3867\gamma^4 - 690\gamma^5 +$

$1202\gamma^6 + 1154\gamma^7 + 445\gamma^8 + 72\gamma^9 + 11\gamma^{10} + 19\gamma^{11})t + \frac{1}{196}(-3308 - 3\gamma + 2574\gamma^2 - 5801\gamma^3 + 339\gamma^4 + 1908\gamma^5 - 936\gamma^6 - 1658\gamma^7 - 1355\gamma^8 - 604\gamma^9 + 45\gamma^{10} - 5\gamma^{11})$ of [6] is irreducible over F . It splits modulo \mathfrak{p} in 3 linear factors and one factor of degree 11. This leads to consider possible factors of degree one (T_2 -bound of 3051420855768 and exponent 15), two (T_2 -bound of 131313006031604 and exponent 17) and three (T_2 -bound of 525251952028508 and exponent 19).

Depending on the estimate for $\text{cond}(M)$ this induces exponent bounds of 23 (24) for linear factors, 25 (27) for degree two and 26 (27) for degree three.

It takes 310 ms to factor f over F using the algorithm of Pohst [6]. The creation of the real lattice (including the LLL-reduction) takes 140 ms, whereas our new method takes 440 ms.

8.3 Factorization 2

In the average case (not totally real), the new method takes advantage of the integer arithmetic and is much faster than the algorithm of Pohst [6]. We also compare our algorithm to a Trager [11] base method which is implemented in KASH [4].

Next we consider the field generated by a root ρ of $x^5 - 3x^4 - 10x^3 - 3x^2 - 15x + 10$. This field has signature (3, 1) and discriminant -2282655415 . As polynomials we choose “random” polynomials, i.e. we generate polynomials where the coefficients (of the coefficients) are bounded by B . We start by multiplying m polynomials of degree n and try to recover them using both our method, Pohst’s method and the Trager [11] based method. The numbers t_1 , t_2 and t_3 in table 1 denote the time for our new method, Pohst’s method and the Trager based method in seconds. All are average times for several different polynomials.

Table 1. Experimental results

n	m	B	t_1	t_2	t_3
5	3	20	0.5	0.8	0.71
7	3	20	0.6	2.0	1.4
10	3	20	1.2	2.7	2.8
15	3	20	1.4	10.6	6.47
5	5	20	1.2	3.3	2.4
10	5	20	3.0	28.2	11.6
5	3	2^{10}	0.4	1.9	1.0

Finally we consider the field generated by a root ρ of $f = x^{10} - 16x^9 + 14x^8 + 2x^7 - 13x^6 + 18x^5 - 8x^4 - 11x^3 + 6x^2 + x - 5$. This field has signature (2, 4), the equation order is already maximal. In table 2 we omit times for Pohst’s method. Because of the size of the involved numbers, the real precision necessary to recover (small) linear factors is enormous. Even to generate $\Delta_{\mathbb{R}}(\mathfrak{a})$ takes more than 30 sec. In this case, we used $\mathfrak{a} := (1073741741R + (577625038 + \rho)R)^{25}$.

Table 2. Experimental results

n	B	t_1	t_2
5	20	1.3	7.2
10	20	3.0	26.0
15	20	3.5	60.1
20	20	4.2	110.1
25	20	5.6	184.3
5	2^{27}	3.6	31.2
10	2^{27}	4.8	113.9
15	2^{27}	8.4	262.9
5	2^{40}	6.4	54.4

8.4 Roots

We consider the field $F := \mathbb{Q}(\sqrt{25601}, \zeta_7)$ which arises during the computation of the Hilbert class field of $\mathbb{Q}(\sqrt{25601})$. One step in the algorithm requires us to take a certain 7th root of a smooth non-integral number γ (smooth meaning the number has only small prime ideal divisors). In this particular example, the number field has degree 12 over \mathbb{Q} , the coefficients of γ have approximately 420 decimal digits and the denominator has 1030 digits (therefore we omit the exact value). Choosing a prime ideal \mathfrak{p} over 11 (of degree 3) we compute $\alpha \bmod \mathfrak{p}^k$ for $k = 2, 4, \dots, 1024$ and using the results of section 5, it takes 12 seconds to compute the root.

Clearing the denominator (i.e. computing the root of $\gamma \times \text{den}(\gamma)^7$) requires us to lift up to \mathfrak{p}^{4096} . This process take 400 seconds.

Clearing only the part of the denominator that is no 7th power (the denominator is $2^{336} \cdot 7^{30} \cdot 41^{560}$, i.e. multiplying with $2^{336} \cdot 7^{28} \cdot 41^{560}$) leaves us with a denominator of 7^2 . Now, using section 5 requires a precision of \mathfrak{p}^{256} and a total runtime of 2.5 seconds.

References

1. B. Beauzamy, *Products of Polynomials and A Priori Estimates for Coefficients in Polynomial Decompositions: A Sharp Result*, J. Symb. Comp. 13, 463–472 (1992).
2. M. Encarnación, *Faster Algorithms for Reconstructing Rationals, Computing Polynomial GCDs, and Factoring Polynomials*, Ph. D. Thesis, Linz 1995.
3. A. K. Lenstra, *Factoring polynomials over algebraic numberfields*, LN in Comp. Sci. 144, 32–39 (1982).
4. M. Daberkow, C. Fieker, J. Klüners, M. E. Pohst, K. Roegner, M. Schörnig, K. Wildanger, *KANT V4*, J. Symb. Comput. 24, 267–283 (1997).
5. A. K. Lenstra, H. W. Lenstra Jr. and L. Lovász, *Factoring Polynomials with Rational Coefficients*, Math. Ann. 261, 515–534 (1982).
6. M. E. Pohst, *Factoring polynomials over global fields*, submitted to J. Symb. Comput. (1999).

7. M. E. Pohst, H. Zassenhaus, *Algorithmic Algebraic Number Theory*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1989.
8. X.-F. Roblot, *Algorithmes de factorisation dans les corps de nombres et applications de la conjecture de Stark à la construction des corps de classes de rayon*, Ph. D. Thesis, Bordeaux 1997.
9. J. Stoer, *Numerische Mathematik 1*, Springer-Lehrbuch, Springer-Verlag, 1993.
10. J. Stoer, R. Bulirsch, *Numerische Mathematik 2*, Springer-Lehrbuch, Springer-Verlag, 1990.
11. B. M. Trager, *Algebraic factoring and rational function integration*, Proceedings of the 1976 Symposium on Symbolic and Algebraic Computation, 219–226, ACM Press 1976.

Dissecting a Sieve to Cut Its Need for Space

William F. Galway

Department of Mathematics, University of Illinois at Urbana-Champaign
1409 West Green Street, Urbana, IL 61801
galway@math.uiuc.edu
<http://www.math.uiuc.edu/~galway>

Abstract. We describe a “dissected” sieving algorithm which enumerates primes in the interval $[x_1, x_2]$, using $O(x_2^{1/3})$ bits of memory and using $O(x_2 - x_1 + x_2^{1/3})$ arithmetic operations on numbers of $O(\ln x_2)$ bits. This algorithm is based on a recent algorithm of Atkin and Bernstein [1], modified using ideas developed by Voronoï for analyzing the Dirichlet divisor problem [20]. We give timing results which show our algorithm has roughly the expected running time.

1 Introduction

The sieve of Eratosthenes, or one of its many variants developed over the last few decades, remains the method of choice for locating primes in an interval $[x_1, x_2]$, provided the interval is sufficiently long. However, these methods appear to suffer from an overhead of $O(x_2^{1/2+o(1)})$ operations as $x_2 \rightarrow \infty$, making sieving inefficient for intervals much shorter than $x_2^{1/2}$. This puts a lower bound on the amount of memory required to sieve efficiently. More precisely, let us say that the interval $[x_1, x_2]$ has length $1+x_2-x_1$. Then, given $B \geq 1$ bits of memory and given $x_1 \leq x_2 < x_1 + B$, we can sieve to find the primes p , $x_1 \leq p \leq x_2$, using $O((x_2-x_1)x_2^{o(1)} + x_2^{1/2+o(1)})$ operations. An interval of arbitrary length, perhaps longer than B , may be divided into several segments of length B , and possibly one more shorter segment. Summing the operation counts to sieve each segment gives a total of $O(x_2^{o(1)}(x_2-x_1)(1+x_2^{1/2}/B) + x_2^{1/2+o(1)})$ operations to sieve the entire interval, and we see that this is inefficient if B is much smaller than $x_2^{1/2}$. A survey of sieving algorithms and their memory requirements is given in a recent paper by Sorenson [17].

When sieving the entire interval from 2 through x , memory requirements are unlikely to impede the calculation because the time to reach an x for which sieving becomes inefficient, say $x \approx 10^{20}$, would be very great. For example, Nicely has been conducting a careful study of the distribution of prime gaps, twin primes, etc., and has taken roughly 7 years to sieve through 10^{15} [13,14]. However, for other applications memory requirements may be a serious limitation. For example, the Lagarias-Odlyzko “analytic” algorithm for computing $\pi(x)$, requires the enumeration of primes in an interval about x , of length $x^{1/2+\varepsilon}$ [11]. Although the Lagarias-Odlyzko algorithm is asymptotically the fastest known

method for computing $\pi(x)$, it is expected that x must be quite large, perhaps as large as 10^{24} or more, before this method becomes faster than other methods such as the extended Meissel-Lehmer method [10,2].

The primality of a given n can be determined using $O(n^\varepsilon)$ arithmetic operations and $O(n^\varepsilon)$ bits of memory, e.g., by using the APRCL or ECPP algorithms [15]. However, the cost per n appears to be significantly greater than the cost for sieving, provided we have enough memory and a sufficiently long interval to sieve. We describe an algorithm which sieves efficiently with significantly smaller memory requirements. This algorithm uses ideas from a recently developed algorithm of Atkin and Bernstein, which are then modified to give a “dissected sieve” that enumerates primes in the interval $[x_1, x_2]$ using $O(x_2^{1/3})$ bits and $O(x_2 - x_1 + x_2^{1/3})$ arithmetic operations on numbers of $O(\ln x_2)$ bits.

2 The Atkin-Bernstein Algorithm

Atkin and Bernstein base their algorithm on classical theorems which relate primality to properties of binary quadratic forms [1]. Theorem 1 below paraphrases their formulation. It uses a different but equivalent condition for Case (a), and is stated so that there is no overlap between the congruence classes considered.

Theorem 1. *Let n be a positive integer, and*

- (a) *if $n \equiv 1 \pmod{4}$ let $\mathcal{R} = \{(u_1, u_2) : u_1 > u_2 > 0\}$, $Q(u_1, u_2) = u_1^2 + u_2^2$;*
- (b) *if $n \equiv 7 \pmod{12}$ let $\mathcal{R} = \{(u_1, u_2) : u_1 > 0, u_2 > 0\}$, $Q(u_1, u_2) = 3u_1^2 + u_2^2$;*
- (c) *if $n \equiv 11 \pmod{12}$ let $\mathcal{R} = \{(u_1, u_2) : u_1 > u_2 > 0\}$, $Q(u_1, u_2) = 3u_1^2 - u_2^2$.*

Let $P(n) = \#\{(u_1, u_2) \in \mathbb{Z}^2 \cap \mathcal{R} : Q(u_1, u_2) = n\}$. Then n is prime if and only if n is squarefree and $P(n)$ is odd.

For any n not divisible by 2 or 3, Algorithm 1 gives a method for determining the primality of a single n in $O(\sqrt{n})$ operations. The method can be made much more efficient when determining the primality of all n , $x_1 \leq n \leq x_2$, provided $x_2 - x_1$ and the available memory are large enough. (Just as the $O(\sqrt{n})$ method of trial division for a single n leads to the much more efficient sieve of Eratosthenes.)

In the following discussion, `pbuf` is a “bit vector” data structure of B bits, indexed by n as $\text{pbuf}[n] \in \{0, 1\}$, and having two components `pbuf.x1` and `pbuf.x2` giving the range of valid indices. We use the convention $\text{pbuf}[n] = 1$ if and only if n is prime. Given $3 < x_1 \leq n \leq x_2 < x_1 + B$, Algorithm 1 below uses Theorem 1 to compute $\text{pbuf}[n] = P(n) \bmod 2$. (Algorithms 1 and 2 are meant to serve as subroutines for sieving segments of length $\leq B$ when sieving an interval of arbitrary length.)

After initializing `pbuf[n]` to zero, $x_1 \leq n \leq x_2$, Algorithm 1 enumerates lattice points $(u_1, u_2) \in \mathcal{R}$ that are bounded between (or lie on) the conic sections $Q(u_1, u_2) = x_1$ and $Q(u_1, u_2) = x_2$, where \mathcal{R} and the matching $Q(u_1, u_2)$ range through the three cases of Theorem 1. For each such point, the corresponding n is complemented when n is in the congruence class appropriate for the quadratic

form. Having computed $\text{pbuf}[n] = P(n) \bmod 2$, the algorithm sieves out numbers with square factors in a final pass.

For a given quadratic form and associated congruence class, the lattice points within the swath $x_1 \leq Q(u_1, u_2) \leq x_2$, $(u_1, u_2) \in \mathcal{R}$, are enumerated by varying u_1 , and then for fixed u_1 incrementing u_2 along a vertical scanline, choosing the starting and ending values of u_2 to avoid points outside the swath. Enumerated points are illustrated in Figure 1 and correspond to the dark points within a swath. (We show all lattice points within a swath, not just those satisfying the corresponding congruence condition.)

Our algorithms are presented in a mixture of mathematical and C-like notations. Assignment is denoted by \leftarrow , with $x++$ as shorthand for $x \leftarrow x + 1$. Assignments may be treated as expressions with values, as in $(n \leftarrow u_1^2 + u_2^2) \leq x_2$, where the value assigned to n is then compared with x_2 . We write $\&\&$ for the conditional conjunction of boolean expressions, so that in the expression `expr1 && expr2`, the second subexpression is evaluated only if `expr1` is TRUE, and the boolean value of the expression is the logical “and” of the two subexpressions. Comments begin with “//” while important conditions or truths are emphasized with **assert** statements.

Algorithm 1 *Given a preallocated bit vector `pbuf`, indexed by n in the range $3 < \text{pbuf.x1} \leq n \leq \text{pbuf.x2}$, this algorithm sets `pbuf[n]` such that upon completion we have `pbuf[n] = 1` if and only if n is prime.*

```

1 SieveSegment( pbuf ) {
2    $x_1 \leftarrow \text{pbuf.x1}; x_2 \leftarrow \text{pbuf.x2}; \text{assert } 3 < x_1 \leq x_2;$ 
3   for ( $n \leftarrow x_1; n \leq x_2; n++$ ) pbuf[n] ← 0;
4   // Case (a)  $n \equiv 1 \pmod{4}$ , handles  $n \bmod 12 \in \{1, 5, 9\}$ .
5   for ( $u_1 \leftarrow \lceil(x_1/2)^{1/2}\rceil; u_1^2 \leq x_2; u_1++$ )
6     for ( $u_2 \leftarrow \lceil \max(0, x_1 - u_1^2)^{1/2} \rceil; u_2 < u_1 \&\& (n \leftarrow u_1^2 + u_2^2) \leq x_2; u_2++$ )
7       if ( $n \bmod 4 = 1$ ) pbuf[n] ← pbuf[n] + 1 mod 2;
8   // Case (b)  $n \equiv 7 \pmod{12}$ .
9   for ( $u_1 \leftarrow 1; 3u_1^2 \leq x_2; u_1++$ )
10    for ( $u_2 \leftarrow \lceil \max(0, x_1 - 3u_1^2)^{1/2} \rceil; u_2 < u_1 \&\& (n \leftarrow 3u_1^2 + u_2^2) \leq x_2; u_2++$ )
11      if ( $n \bmod 12 = 7$ ) pbuf[n] ← pbuf[n] + 1 mod 2;
12   // Case (c)  $n \equiv 11 \pmod{12}$ .
13   for ( $u_1 \leftarrow \lceil(x_1/3)^{1/2}\rceil; 2u_1^2 \leq x_2; u_1++$ )
14     for ( $u_2 \leftarrow \lceil \max(0, 3u_1^2 - x_2)^{1/2} \rceil; u_2 < u_1 \&\& (n \leftarrow 3u_1^2 - u_2^2) \geq x_1; u_2++$ )
15       if ( $n \bmod 12 = 11$ ) pbuf[n] ← pbuf[n] + 1 mod 2;
16   // Sieve out numbers with square factors .
17   for ( $q \leftarrow 3; q^2 \leq x_2; q++$ )
18     for ( $m \leftarrow \lceil x_1/q^2 \rceil; mq^2 \leq x_2; m++$ )
19       pbuf[mq^2] ← 0;
```

The presentation of Algorithm 1 given here is quite different from that of Atkin and Bernstein. They avoid enumerating points and allocating storage for n divisible by small primes, and they use difference equations satisfied by the

quadratic forms to reduce the number of multiplications and square root operations required. However, both versions have the same basic complexity. Up to O -constants, the operation count for Algorithm 1 is the sum of the number of lattice points enumerated, the number of scanlines required to find them, and the number of operations required by the squarefree sieve of lines 18–20.

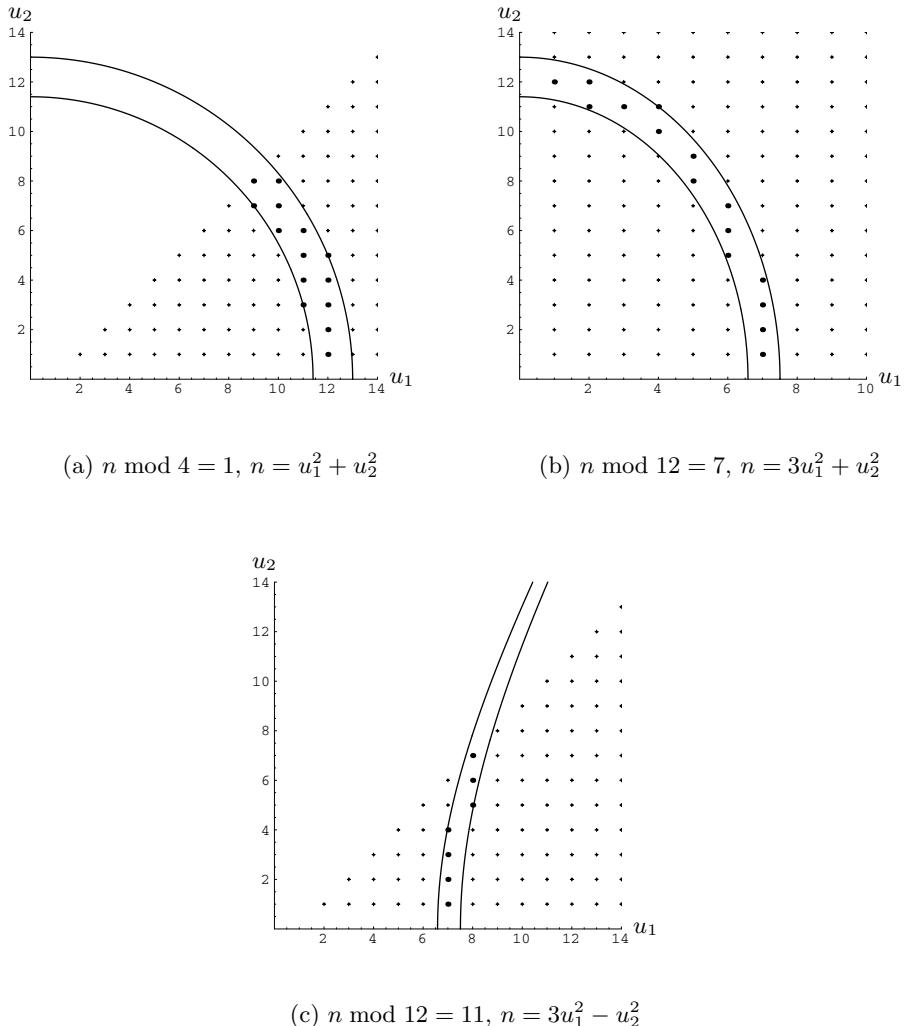


Fig. 1. The three cases of Theorem 1 and of Algorithms 1 and 2

It is easy to show that the squarefree sieve requires $O(x_2 - x_1 + x_2^{1/2})$ operations, and that there are $O(x_2^{1/2})$ scanlines. The number of lattice points is closely related to the area of the swath they lie in—the number differing from the area by an amount which is $O(x_2^{1/2})$. For each of the three cases, the area is $O(x_2 - x_1)$. We conclude that the number of lattice points, and the total operation count for Algorithm 1, is $O(x_2 - x_1 + x_2^{1/2})$. When Algorithm 1 is used to sieve segments of length $\leq B$ in an interval of arbitrary length, the operation count for the entire interval is $O((x_2 - x_1)(1 + x_2^{1/2}/B) + x_2^{1/2})$. Thus, sieving with Algorithm 1 is inefficient if B is much less than $x_2^{1/2}$. (Atkin and Bernstein describe a further modification which reduces the operation count by a factor of $\ln \ln x_2$, at the cost of slightly greater memory requirements.)

We may refine the bound of $O(x_2^{1/2})$ for the difference between the area of a swath and the number of points within it. The question of the minimal upper bound is closely related to the “circle problem”, which is concerned with estimating the difference between the number of lattice points within a circle of radius \sqrt{x} and its area. By a result of van der Corput [19] it follows that for each case of Algorithm 1 the number of points enumerated is $O(x_2 - x_1 + x_2^{1/3})$. Van der Corput’s result generalizes earlier work of Voronoï on the Dirichlet divisor problem [20], and of Sierpiński on the circle problem [16]. (See Section 5 of this paper and the discussion following Theorem 2.4.2 in [7].) Furthermore, the key idea behind these results lets us reduce the number of scanlines needed to enumerate the points, giving our dissected sieve described below.

3 Dissecting the Atkin-Bernstein Algorithm

To reduce the number of scanlines, we dissect the swath into pieces, and then scan each piece in a direction roughly tangent to the boundary curves (see Figures 2 and 3). We choose tangents with slopes defined by a Farey sequence of order r , and then use corresponding “cuts” to separate the pieces. The optimal choice for r is discussed below in Subsection 3.3.

3.1 Notation and Background Material

Before giving details of the dissection, we introduce some notation and state without proofs some properties of quadratic forms and of Farey sequences.

We use vector notation, with vectors denoted by lowercase boldface letters, and matrices by uppercase boldface letters. The transpose of \mathbf{A} is written \mathbf{A}^t .

Definition 1. Given $\mathbf{u} = [u_1 \ u_2]^t$ and a symmetric matrix \mathbf{A} , let

$$Q_{\mathbf{A}}(\mathbf{u}) = \mathbf{u}^t \mathbf{A} \mathbf{u} = a_1 u_1^2 + a_2 u_1 u_2 + a_3 u_2^2,$$

where \mathbf{A} and the coefficients a_j are related by

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2/2 \\ a_2/2 & a_3 \end{bmatrix}.$$

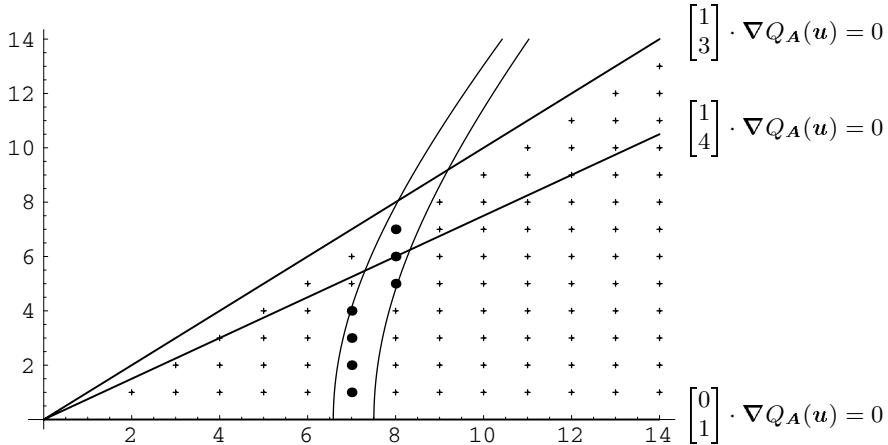


Fig. 2. A dissection for $Q_A(\mathbf{u}) = 3u_1^2 - u_2^2$, using three cuts

Lemma 1. Given a symmetric matrix \mathbf{A} , then

$$Q_{\mathbf{A}}(\rho \mathbf{u}) = \rho^2 Q_{\mathbf{A}}(\mathbf{u}), \quad (1)$$

$$Q_{\mathbf{A}}(\mathbf{u} + \mathbf{v}) = Q_{\mathbf{A}}(\mathbf{u}) + Q_{\mathbf{A}}(\mathbf{v}) + 2\mathbf{v}^t \mathbf{A}\mathbf{u}, \quad (2)$$

$$\nabla Q_{\mathbf{A}}(\mathbf{u}) = 2\mathbf{A}\mathbf{u}. \quad (3)$$

Definition 2. Given a quadratic form $Q_{\mathbf{A}}(\mathbf{u})$ and vector τ , let the *cutting line for τ* , or τ -cut, be the set of points \mathbf{u} at which the gradient $\nabla Q_{\mathbf{A}}(\mathbf{u})$ is perpendicular to τ , i.e.,

$$\{\mathbf{u} : \tau \cdot \nabla Q_{\mathbf{A}}(\mathbf{u}) = 0\} = \{\mathbf{u} : \kappa \cdot \mathbf{u} = 0\}$$

where $\kappa = \mathbf{A}\tau$. We call κ the *coefficient vector* for the cut.

The τ -cut is a line passing through the origin. It depends on the quadratic form $Q_{\mathbf{A}}(\mathbf{u})$ as well as on τ ; the quadratic form should be clear from context. For a given x , the curve $Q_{\mathbf{A}}(\mathbf{u}) = x$ is parallel to τ where the curve intersects the τ -cut. In fact, given a line $\kappa \cdot \mathbf{u} = 0$ the vector $\tau = \mathbf{A}^{-1}\kappa$ is the associated tangent vector at the intersection of $\kappa \cdot \mathbf{u} = 0$, $Q_{\mathbf{A}}(\mathbf{u}) = x$.

We define \mathcal{F}_r , the Farey sequence of order r , to be the ascending sequence of reduced fractions β/α between 0 and 1, with denominators bounded by r :

$$\mathcal{F}_r = \{\beta/\alpha : \gcd(\alpha, \beta) = 1, 0 \leq \beta \leq \alpha \leq r\}.$$

Writing $\beta/\alpha < \beta'/\alpha'$ for consecutive elements of \mathcal{F}_r , we have

$$\alpha\beta' - \alpha'\beta = 1 \quad (\text{Theorem 28 in Hardy and Wright [6].}) \quad (4)$$

Lemma 2 below is from *The Art of Computer Programming* [9, Exercise 1.3.2.18], and follows from [6, §3.4].

Lemma 2 (Recursion for Farey Fractions). *Let $\beta_0/\alpha_0, \beta_1/\alpha_1, \dots$ denote the Farey sequence of order r . Then,*

$$\begin{aligned}\beta_0 &= 0, \quad \alpha_0 = 1; \quad \beta_1 = 1, \quad \alpha_1 = r; \\ \beta_{k+2} &= \lfloor (r + \alpha_k)/\alpha_{k+1} \rfloor \beta_{k+1} - \beta_k; \\ \alpha_{k+2} &= \lfloor (r + \alpha_k)/\alpha_{k+1} \rfloor \alpha_{k+1} - \alpha_k.\end{aligned}$$

3.2 The Dissection Algorithm

To dissect a swath, Algorithm 2 below uses a sequence of tangent vectors derived from $\beta/\alpha \in \mathcal{F}_r$, of the form $\tau = [-\beta \ \alpha]^t$ in Cases (a) and (b), and $\tau = [\beta \ \alpha]^t$ in Case (c) of Theorem 1, so that in all cases the corresponding cutting lines, $\kappa \cdot \mathbf{u} = 0$, run through the upper-right quadrant. For Case (c) we terminate the sequence upon reaching a cut of slope 1, i.e. $\beta/\alpha = 1/3$, so we must choose $r \geq 3$. To dissect the swath over the entire quadrant in Case (b), we swap the roles of u_1 and u_2 and then perform a similar dissection for $Q_A(\mathbf{u}) = u_1^2 + 3u_2^2$.

The *piece* corresponding to consecutive tangents τ and τ' is the set of points \mathbf{u} with $x_1 \leq Q_A(\mathbf{u}) \leq x_2$ and with \mathbf{u} between the τ -cut and the τ' -cut. We exclude points on the τ -cut and include points on the τ' -cut. In Cases (a) and (c) of Theorem 1 the included points lying on the very last τ' -cut, i.e., on the line $u_2 = u_1$, lie outside the corresponding \mathcal{R} . Similarly, in Case (b) the points on the line $u_2 = 3u_1$ are counted twice. Although this gives an incorrect calculation of $P(n) \bmod 2$ for n corresponding to these points, the end result is still correct because such n have square factors and are sieved out in the final pass.

Algorithm 2 controls initialization, dissection into pieces, and calling the squarefree sieve routine; while Algorithm 3 (`ScanPiece`) scans a single piece, and Algorithm 4 (`SquareFreeSieve`) does the final sieving to eliminate square factors. We discuss each algorithm in turn.

Algorithm 2 (SieveSegment: Dissected Sieve of Order r) *Given $r \geq 3$ and a preallocated bit vector pbuf as in Algorithm 1, this algorithm sets pbuf[n] such that upon completion we have pbuf[n] = 1 if and only if n is prime.*

```

1 SieveSegment(r, pbuf) {
2   assert 3 < pbuf.x1 ≤ pbuf.x2;
3   assert r ≥ 3;
4   β ← 0; α ← 1;
5   β' ← 1; α' ← r;
6   while (TRUE) {
7     // Case (a) n ≡ 1 (mod 4), handles n mod 12 ∈ {1, 5, 9}.
8     ScanPiece(1, 4, [1 0], [-β α]^t, [-β' α']^t, pbuf);
9     // Case (b) n ≡ 7 (mod 12).
10    ScanPiece(7, 12, [3 0], [-β α]^t, [-β' α']^t, pbuf);
11    ScanPiece(7, 12, [1 0], [-β α]^t, [-β' α']^t, pbuf);
12    if (3β' ≤ α') { // Case (c) n ≡ 11 (mod 12).}
```

```

13 ScanPiece (11,12,[ $\begin{smallmatrix} 3 & 0 \\ 0 & -1 \end{smallmatrix}$ ],[ $\beta \alpha$ ] $^t$ ,[ $\beta' \alpha'$ ] $^t$ , pbuf);}
14
15 if ( $\beta' = \alpha'$ )
16   break ;
17 // Advance to next Farey fraction of order r.
18  $k \leftarrow \lfloor (r + \alpha) / \alpha' \rfloor$ ;
19  $\{\beta, \beta'\} \leftarrow \{\beta', k\beta' - \beta\}$ ;
20  $\{\alpha, \alpha'\} \leftarrow \{\alpha', k\alpha' - \alpha\}$ ;
21 }
22 // Sieve out square factors, using Algorithm ~4 below.
23 SquareFreeSieve (pbuf);}

```

To scan a piece bounded by the cuts for tangents τ, τ' , Algorithm 3 transforms to another coordinate system, or “ v -space”. The coordinates are related by $\mathbf{u} = \mathbf{T}\mathbf{v}$, with $\mathbf{T} = [\tau \ \tau']$, so the map $\mathbf{v} = \mathbf{T}^{-1}\mathbf{u}$ sends τ to the unit horizontal vector and τ' to the unit vertical vector. By the method used to construct τ and τ' from Farey fractions, Equation (4) implies $\det(\mathbf{T}) = \pm 1$, so the mapping is area-preserving and gives a one-one map between points in \mathbb{Z}^2 (see Figure 3).

Working in v -space, Algorithm 3 scans both horizontally and vertically, shifting between horizontal and vertical at the image of the “median-line” defined in Figure 3(a). We compute $Q_{\mathbf{A}}(\mathbf{u})$ using

$$Q_{\mathbf{A}}(\mathbf{u}) = Q_{\mathbf{B}}(\mathbf{v}) = b_1 v_1^2 + b_2 v_1 v_2 + b_3 v_2^2$$

with

$$\mathbf{B} = \mathbf{T}^t \mathbf{A} \mathbf{T} = \begin{bmatrix} Q_{\mathbf{A}}(\tau) & \kappa \cdot \tau' \\ \kappa \cdot \tau' & Q_{\mathbf{A}}(\tau') \end{bmatrix} = \begin{bmatrix} b_1 & b_2/2 \\ b_2/2 & b_3 \end{bmatrix}. \quad (5)$$

Writing $\mathbf{B} = [\mathbf{b} \ \mathbf{b}']$, the cutting lines and their median-line have images in v -space satisfying

$$\begin{aligned} \mathbf{b} \cdot \mathbf{v} &= 0 && \text{for the } \tau\text{-cut} \\ \mathbf{b}' \cdot \mathbf{v} &= 0 && \text{for the } \tau'\text{-cut} \\ (\mathbf{b} + \mathbf{b}') \cdot \mathbf{v} &= 0 && \text{for the median-line.} \end{aligned}$$

To bound the range of the scanlines, we use three “crossing points” illustrated in Figure 3(b), and defined in terms of “normalized crossing points” $\mathbf{c}, \mathbf{c}', \mathbf{c}''$ given by

$$\begin{aligned} \mathbf{c} &= |\det^{-1}(\mathbf{T})|(a_1 a_3 b_1)^{-1/2} [b_2/2 \quad -b_1]^t \\ \mathbf{c}' &= |\det^{-1}(\mathbf{T})|(a_1 a_3 b_3)^{-1/2} [b_3 \quad -b_2/2]^t \\ \mathbf{c}'' &= |\det^{-1}(\mathbf{T})|(a_1 a_3 (b_1 + b_2 + b_3))^{-1/2} [(b_2/2 + b_3) \quad (-b_1 - b_2/2)]^t. \end{aligned}$$

(We allow for the possibility that $\det(\mathbf{T}) \neq \pm 1$ in future implementations of Algorithm 3.) For diagonal forms $Q_{\mathbf{A}}(\mathbf{u})$, these points are at intersections between

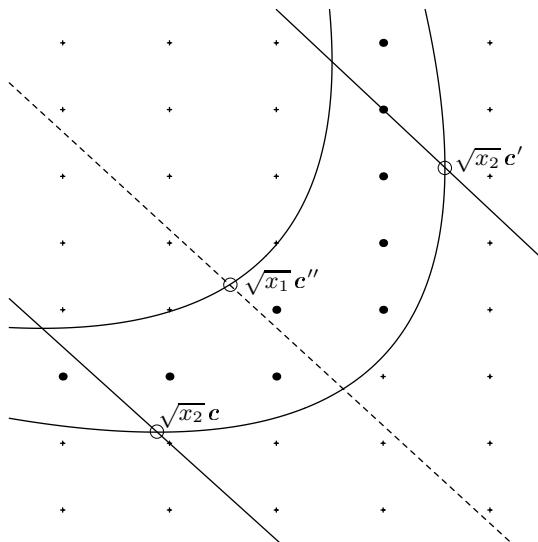
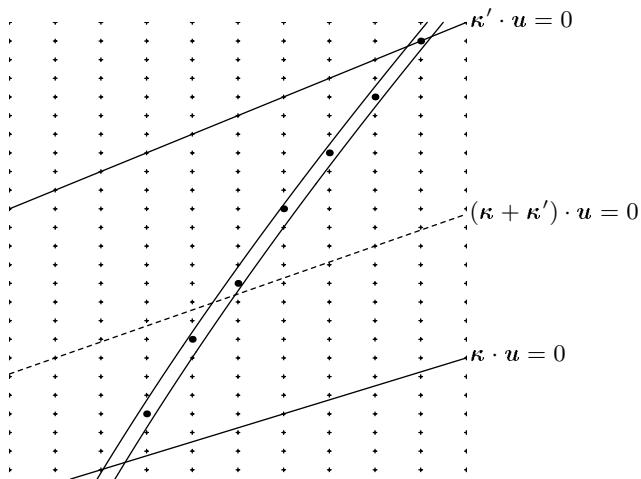


Fig. 3. A piece of a dissection, in two coordinate systems

$Q_B(\mathbf{v}) = 1$ and the images in \mathbf{v} -space of the τ -cut, τ' -cut, and mediant-line, respectively. By Equation (1), we see that $\sqrt{x} \mathbf{c}$ lies at an intersection between $Q_B(\mathbf{v}) = x$ and the image of the τ -cut, and similarly for \mathbf{c}' , \mathbf{c}'' . Given $x_1 \geq 0$, $x_2 \geq 0$, we must have $\det(\mathbf{T})(x_2 - x_1) > 0$ for the point $\sqrt{x_1} \mathbf{c}''$ to lie above $\sqrt{x_2} \mathbf{c}$ and to the left of $\sqrt{x_2} \mathbf{c}'$. In order to maintain this relationship between the crossing points, Algorithm 3 swaps x_1 and x_2 when $\det(\mathbf{T}) < 0$.

Making use of the relationships described above, Algorithm 3 proceeds in much the same way as Algorithm 1, although finding the bounding points for a scanline is more complicated because of the greater generality of the quadratics to be solved, and because both ends of the scanline may be bounded by either a conic or by a line.

Algorithm 3 (ScanPiece: Process Lattice Points within a Piece)

This routine enumerates all lattice points $\mathbf{u} \in \mathbb{Z} \times \mathbb{Z}$ lying within the piece corresponding to τ and τ' . Letting $n = Q_A(\mathbf{u})$ for each \mathbf{u} enumerated, pbuf[n] is complemented if $n \bmod m = k$.

```

1 ScanPiece (  $k, m, A, \tau, \tau', \text{pbuf}$  ) {
2    $\mathbf{T} \leftarrow [\tau \ \tau']$ ;
3   if (  $\det(\mathbf{T}) > 0$  )
4      $x_1 \leftarrow \text{pbuf}.x1$ ;  $x_2 \leftarrow \text{pbuf}.x2$ ;
5   else
6      $x_1 \leftarrow \text{pbuf}.x2$ ;  $x_2 \leftarrow \text{pbuf}.x1$ ;
7    $b_1 \leftarrow Q_A(\tau)$ ;  $b_2 \leftarrow \tau'^t A \tau$ ;  $b_3 \leftarrow Q_A(\tau')$ ;  $\mathbf{B} \leftarrow \begin{bmatrix} b_1 & b_2/2 \\ b_2/2 & b_3 \end{bmatrix}$ ;
8    $\mathbf{c} \leftarrow |\det^{-1}(\mathbf{T})|(a_1 a_3 b_1)^{-1/2} [b_2/2 \ - b_1]^t$ ;
9    $\mathbf{c}' \leftarrow |\det^{-1}(\mathbf{T})|(a_1 a_3 b_3)^{-1/2} [b_3 \ - b_2/2]^t$ ;
10   $\mathbf{c}'' \leftarrow |\det^{-1}(\mathbf{T})|(a_1 a_3 (b_1 + b_2 + b_3))^{-1/2} [(b_2/2 + b_3) \ (-b_1 - b_2/2)]^t$ ;
11 // Scan the half-piece below, or on, the mediant-line
12 // (horizontal scanlines).
13 for (  $v_2 \leftarrow \lceil [0 \ 1] \sqrt{x_2} \mathbf{c} \rceil$ ;  $v_2 \leq \lfloor [0 \ 1] \sqrt{x_1} \mathbf{c}'' \rfloor$ ;  $v_2++$  ) {
14   // d = discriminant of quadratic, or zero.
15    $d \leftarrow \max(0, b_2^2 v_2^2 - 4b_1(b_3 v_2^2 - x_1))$ ;
16   // v_start ∈ ℝ, v_stop ∈ ℝ give limits for the scanline.
17    $v_{\text{start}} \leftarrow (-b_2 v_2 + \sqrt{d})/(2b_1)$ ;
18    $d \leftarrow b_2^2 v_2^2 - 4b_1(b_3 v_2^2 - x_2)$ ; assert  $d \geq 0$ ;
19    $v_{\text{stop}} \leftarrow \min(-(2b_3 + b_2)v_2/(2b_1 + b_2), (-b_2 v_2 + \sqrt{d})/(2b_1))$ ;
20   for (  $v_1 \leftarrow 1 + \lfloor v_{\text{start}} \rfloor$ ;  $v_1 \leq v_{\text{stop}}$ ;  $v_1++$  ) {
21      $n \leftarrow Q_B([v_1 \ v_2]^t)$ ;
22     if (  $n \bmod m = k$  )  $\text{pbuf}[n] \leftarrow \text{pbuf}[n] + 1 \bmod 2$ ; }
23 // Scan the half-piece above, and off, the mediant-line
24 // (vertical scanlines).
25 for (  $v_1 \leftarrow \lceil [1 \ 0] \sqrt{x_1} \mathbf{c}'' \rceil$ ;  $v_1 \leq \lfloor [1 \ 0] \sqrt{x_2} \mathbf{c}' \rfloor$ ;  $v_1++$  ) {
26    $d \leftarrow b_2^2 v_1^2 - 4b_3(b_1 v_1^2 - x_2)$ ; assert  $d \geq 0$ ;
27    $v_{\text{start}} \leftarrow \max(-(2b_1 + b_2)v_1/(2b_3 + b_2), (-b_2 v_1 - \sqrt{d})/(2b_3))$ ;
28    $d \leftarrow \max(0, b_2^2 v_1^2 - 4b_3(b_2 v_1^2 - x_1))$ ;
29    $v_{\text{stop}} \leftarrow (-b_2 v_1 - \sqrt{d})/(2b_3)$ ;

```

```

30   for (  $v_2 \leftarrow 1 + \lfloor v_{\text{start}} \rfloor$ ;  $v_2 \leq v_{\text{stop}}$ ;  $v_2++$ ) {
31      $n \leftarrow Q_B([v_1 \ v_2]^t)$ ;
32     if ( $n \bmod m = k$ ) pbuf[n]  $\leftarrow$  pbuf[n] + 1 mod 2; } }
33 }
```

Algorithm 4 is similar to the corresponding code of lines 18–20 in Algorithm 1, but uses the “Dirichlet hyperbola method” [18, Chapter I.3, §3.2] to reduce the operation count from $O(x_2 - x_1 + x_2^{1/2})$ to $O(x_2 - x_1 + x_2^{1/3})$.

Algorithm 4 (SquareFreeSieve: Sieve out Square Factors)

This routine sets pbuf[n] = 0 for each n of the form n = mq², q > 1.

```

1 SquareFreeSieve ( pbuf ) {
2    $x_1 \leftarrow$  pbuf.x1;  $x_2 \leftarrow$  pbuf.x2;
3   for ( $q \leftarrow 3$ ;  $q \leq x_2^{1/3}$ ;  $q++$ )
4     for ( $m \leftarrow \lceil x_1/q^2 \rceil$ ;  $mq^2 \leq x_2$ ;  $m++$ )
5       pbuf[mq2]  $\leftarrow$  0;
6     for ( $m \leftarrow 1$ ;  $m \leq x_2^{1/3}$ ;  $m++$ )
7       for ( $q \leftarrow \max(3, \lceil (x_1/m)^{1/2} \rceil)$ ;  $mq^2 \leq x_2$ ;  $q++$ )
8         pbuf[mq2]  $\leftarrow$  0; }
```

3.3 Timing Analysis and Optimal Order of Dissection

Up to O -constants, the operation count for Algorithm 2 is the sum of the numbers of points, scanlines, pieces, and operations required by Algorithm 4. As mentioned above, Algorithm 4 requires $O(x_2 - x_1 + x_2^{1/3})$ operations, and, by the work of van der Corput [19], the number of points is of the same order.

For a given r , the number of pieces is $O(r^2)$ since the number of Farey fractions of order r is $3r^2/\pi^2 + O(r \ln r)$. (See Theorem 330 in Hardy and Wright [6].) The number of pieces is dominated by the number of scanlines, which is shown in my thesis [5] to be $O(r^3(x_2 - x_1)x_2^{-1/2} + r^2 + x_2^{1/2}/r)$. To roughly minimize the number of scanlines, we assume $x_2 > x_1$ and choose r to balance the r^3 and $1/r$ terms in this bound. This gives $r \asymp (x_2/(x_2 - x_1))^{1/4}$, and the number of scanlines is

$$O(r^2 + x_2^{1/2}/r) = O(x_2^{1/2}(x_2 - x_1)^{-1/2} + x_2^{1/4}(x_2 - x_1)^{1/4}).$$

The second term dominates provided $x_2 - x_1 \gg x_2^{1/3}$, in which case the number of scanlines is $O(x_2^{1/4}(x_2 - x_1)^{1/4})$.

Totaling these counts, we find, with $r \asymp (x_2/(x_2 - x_1))^{1/4}$, that Algorithm 2 requires $O(x_2 - x_1 + x_2^{1/3})$ operations, provided $x_2 - x_1 \gg x_2^{1/3}$. The same bound holds, with $r \asymp x_2^{1/6}$, when $x_2 - x_1 \ll x_2^{1/3}$. When Algorithm 2 is used to sieve segments of length $\leq B$ in an interval of arbitrary length, using the appropriate value of r for each segment, the operation count for the entire interval is $O((x_2 - x_1)(1 + x_2^{1/3}/B) + x_2^{1/3})$.

3.4 Implementation Notes

The dissected sieve has been implemented as a program `dsieve`. Although this implementation is rudimentary, there are some optimizations. We reduce the size of the numbers used in Algorithm 3 (`ScanPiece`) by working both in \mathbf{u} -space and in a coordinate system in which v -space is re-centered around the lattice point given by rounding the components of $\sqrt{x_1} \mathbf{c}''$ to their nearest integers. We also get smaller numbers by computing $Q_{\mathbf{A}}(\mathbf{u}) - x_1$, rather than $Q_{\mathbf{A}}(\mathbf{u})$ directly. As in the Atkin-Bernstein paper, `ScanPiece` uses Equation (2) to reduce the number of multiplications needed to update $Q_{\mathbf{A}}(\mathbf{u}) - x_1$ as \mathbf{u} varies. It reduces the number of square root operations by monitoring the values of $Q_{\mathbf{A}}(\mathbf{u}) - x_1$, $\kappa \cdot \mathbf{u}$, and $\kappa' \cdot \mathbf{u}$, to determine whether a point lies within a piece, and to decide when to move to the next scanline.

Algorithm 4 (`SquareFreeSieve`) was modified to sieve only the odd numbers, and an additional parameter was added to control the point at which it makes the transition from the loop of lines 4–6 to the loop of lines 7–9. For `dsieve`, the optimal transition point was found experimentally to be near $q = 1.5x_2^{1/3}$. We also found experimentally, for a wide range of values of x_2 and B , that setting $r = \lfloor 0.5 + 0.7(x_2/B)^{1/4} \rfloor$ approximately minimizes both the number of scanlines and the operation count. It should be noted that the constants 1.5 and 0.7 are certainly dependent on the details of our implementation, and may also be machine dependent.

3.5 Possible Improvements

There are several ways in which `dsieve` could be improved. For simplicity in coding and analysis, we used a single Farey dissection. It may be more efficient to use three Farey dissections, each of an order chosen to optimize the corresponding case in Theorem 1. Also, “Farey-like” sequences tailored to each quadratic form may be more efficient—Sierpiński used a sequence of the form

$$\{\beta/\alpha : \gcd(\alpha, \beta) = 1, \alpha^2 + \beta^2 \leq r^2\}$$

in his work on the circle problem [16]. Furthermore, besides the three pairs of congruence classes and quadratic forms used in Theorem 1, there are other choices—see the paper of Atkin and Bernstein for one example [1]. It seems to be an open question as to how to determine an optimal set of forms.

Currently, `dsieve` enumerates too many points. Although it does not allocate storage for even indices, it does enumerate all points within a swath, including points yielding $Q_{\mathbf{A}}(\mathbf{u}) \equiv 0 \pmod{2}$. Reducing the number of points enumerated, and avoiding the costly test “ $n \bmod m = k$ ” used in Algorithm 3 (`ScanPiece`) could improve the speed. However, this will not reduce the number of scanlines enumerated—which may dominate the operation count when $x_2 - x_1$ is small enough with respect to $x_2^{1/3}$. Also, $\mathbf{T} \pmod{m}$ determines the periodic pattern of remainders taken by $n = Q_{\mathbf{A}}(\mathbf{u}) \pmod{m}$ as \mathbf{u} moves along a scanline. Since the congruence class of $\mathbf{T} = [\tau \ \tau']$ changes irregularly between calls to `ScanPiece`, it may be preferable to restrict \mathbf{T} to a limited set of congruence classes.

We have not rigorously analyzed the size of numbers used by Algorithms 2, 3, and 4. The implementation `dsieve` uses a mixture of 32-bit and 64-bit integers, and 64-bit floating point numbers, and becomes unreliable near 10^{18} .

4 Timing Comparisons

Bernstein has implemented the Atkin-Bernstein algorithm and posted it on the web as a package of routines, “`primegen 0.97`”; see their paper for the URL [1]. Tables 1 and 2 show running times for `primegen` and for `dsieve`. Both programs were compiled to run on SUN SPARC computers, using the Gnu C-compiler (`gcc`) version 2.8.1, with compilation options `-O3 -mcpu=v8`. Times were measured on a 300 MHz UltraSPARC 5/10 with 64 megabytes of “main” memory. In addition, it has roughly 16 kilobytes of “level-1” cache memory—very fast compared to main memory—and 512 kilobytes of somewhat slower level-2 cache. (The amounts and speeds of cache were estimated using a C implementation of the `mem1d` program given in [3, Appendix E].)

x_1	time (seconds)			
	$B \approx 2.05 \cdot 10^6$	$B = 2^{24}$	$B = 2^{26}$	$B = 2^{28}$
10^9	20	27	73	156
10^{10}	26	27	75	194
10^{11}	49	30	72	204
10^{12}	121	38	72	204
10^{13}	340	67	75	207
10^{14}	1035	157	96	215
10^{15}	3231	438	174	241
10^{16}	11716	1527	491	362
10^{17}	74909	9142	2891	1313

Table 1. Time for `primegen` to count primes in the interval $[x_1, x_1 + 10^9]$, using B bits of memory

The program `primes.c` provided in the `primegen` package was modified to print the count of primes in an interval $[x_1, x_2]$, and was run to count primes in several intervals. These counts were compared with those found by `dsieve`, and by a third program based on Robert Bennion’s “hopping sieve” [4]. Although Bernstein warns that the `primegen` code is not valid past $x = 10^{15}$, all programs returned the same counts except for the interval $[10^{17}, 10^{17} + 10^9]$, where `primegen` counts $10^{17} + 111377247 = 7 \cdot 119522861^2$ and $10^{17} + 158245891 = 11 \cdot 95346259^2$ as primes.

The buffer size used by `primegen` (B , in our notation) is set at compile time. Table 1 shows running times for `primegen` to count primes in the interval $[x_1, x_2 = x_1 + 10^9]$ for several combinations of B and x_1 . In Bernstein’s installation instructions he suggests choosing B so that data used by the inner loop of

the algorithm resides in level-1 cache. (The inner loop treats $n = 60k + d$ for fixed d , where d takes one of the 16 values relatively prime to 60.) For the UltraSPARC computer, we used the suggested value of $B = 16 \cdot 128128 \approx 2.05 \cdot 10^6$.

To avoid having a runtime that became linear in x_1 for very large x_1 (linear with a small O -constant), we modified the routine `primegen_skipto` to use division instead of repeated subtraction in its calculation of a quotient.

As well as showing that `primegen` slows as $\sqrt{x_2}$ grows larger than B , Table 1 illustrates that the operation count may only roughly predict the running time on a computer with cache memory. Increasing B reduces the operation count for sieving an interval, but also increases the chance that memory references will miss the level-1 cache. This slowdown as the locality of memory references decreases can be striking. On the computer used for these tests, widely scattered references to “main” memory were measured to be roughly 20 times slower than references to level-1 cache. (An informative discussion of cache memory is given in [3, Chapter 3].)

x_1	$B \approx 10x_2^{1/3}$			$B \approx x_2^{1/2}$				
	B	r	time (sec.)	B	r	time (sec.)		
		sqfree	total		sqfree	total		
10^9	$1.26 \cdot 10^4$	14	51	391	$4.47 \cdot 10^4$	10	25	333
10^{10}	$2.22 \cdot 10^4$	19	55	402	$1.05 \cdot 10^5$	13	23	329
10^{11}	$4.66 \cdot 10^4$	27	57	403	$3.18 \cdot 10^5$	17	22	334
10^{12}	$1.00 \cdot 10^5$	39	59	407	$1.00 \cdot 10^6$	22	20	334
10^{13}	$2.15 \cdot 10^5$	59	59	416	$3.16 \cdot 10^6$	30	18	339
10^{14}	$4.64 \cdot 10^5$	86	61	423	$1.00 \cdot 10^7$	39	20	414
10^{15}	$1.00 \cdot 10^6$	125	61	428	$3.16 \cdot 10^7$	52	20	453
10^{16}	$2.15 \cdot 10^6$	183	61	437	$1.00 \cdot 10^8$	70	19	456
10^{17}	$4.64 \cdot 10^6$	268	63	465	$3.16 \cdot 10^8$	93	19	466

Table 2. Time for `dsieve` to count primes in the interval $[x_1, x_2 = x_1 + 10^9]$, using $\approx B$ bits of memory and a Farey dissection of order $r \approx 0.7(x_2/B)^{1/4}$. The “sqfree” column gives the time required to sieve out square factors

Table 2 shows running times for `dsieve` to count primes in the interval $[x_1, x_2 = x_1 + 10^9]$, using two different values of B , with B depending on x_2 . The entries with $B \approx 10x_2^{1/3}$ illustrate the running time when using a “small” amount of memory, while the entries with $B \approx x_2^{1/2}$ show the running time with memory usage comparable to that needed for efficient operation of previously known sieves. In both cases and for all values of x_1 , after computing `pbuf` it took roughly 18 seconds to count the primes (1-bits in `pbuf`). As expected, the running time does not greatly increase as x_1 increases. The slowdown for larger x_1 is presumably due in part to decreasing locality of reference, although more detailed statistics on operation counts should be collected in order to better understand these results.

5 Miscellaneous Remarks

The ideas of this paper can also be used with $Q_{\mathbf{A}}(\mathbf{u}) = u_1 u_2$, giving a dissected “Eratosthenes-like” sieve. This corresponds to the Dirichlet divisor problem, which is concerned with estimating the number, $D(x)$, of lattice points within the hyperbola $u_1 u_2 \leq x$, $u_1 > 0$, $u_2 > 0$. Voronoï [20] used a dissection based on the Farey-like sequence

$$\{\beta/\alpha : \gcd(\alpha, \beta) = 1, \alpha\beta \leq t\},$$

with $t = x^{1/3}$, to show that $D(x) = x \ln(x) + (2\gamma - 1)x + O(x^{1/3} \ln x)$, where $\gamma \approx 0.5772\dots$ is Euler’s constant. His result was an improvement of an earlier result of Dirichlet, who used the “hyperbola method” to get an error term of $O(x^{1/2})$ instead of $O(x^{1/3+\varepsilon})$ for the approximation of $D(x)$. Voronoï’s result for the Dirichlet divisor problem suggests that a dissected Eratosthenes-like sieve would require $O(x_2^{1/3+\varepsilon})$ bits and $O(x_2^\varepsilon(x_2 - x_1 + x_2^{1/3}))$ operations to sieve the interval $[x_1, x_2]$.

We can also use dissection to improve some factoring algorithms. For example, trial division searches for a solution to $n = Q_{\mathbf{A}}(\mathbf{u}) = u_1 u_2$, and dissection would reduce the operation count to $O(n^{1/3+\varepsilon})$. Dissection would similarly reduce the number of operations needed to solve the quadratics used by D. H. and Emma Lehmer [12] for factoring.

Sierpiński’s $O(x^{1/3})$ bound has since been improved to an $O(x^{35/108+\varepsilon})$ bound for the difference between the number of lattice points within a circle of radius \sqrt{x} and its area [8, § 13.8], and the conjectured bound is $O(x^{1/4+\varepsilon})$. The improved bound was proven using analytic techniques, and it is not clear if these techniques could be applied to making sieving more efficient. However, the result does raise the intriguing speculation that we may be able to sieve efficiently using significantly less than $O(x_2^{1/3})$ bits of memory.

Acknowledgments

Dan Bernstein first brought my attention to the Atkin-Bernstein sieving algorithm. Thomas Nicely provided helpful information about his sieving project. Jared Bronski, at the University of Illinois Urbana-Champaign, made his SUN workstation available for the timing benchmarks.

References

1. A. O. L. Atkin and D. J. Bernstein, *Prime sieves using binary quadratic forms*, Dept. of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 60607-7045. Preprint available at <http://pobox.com/~djb/papers/primesieves.dvi>, 1999.
2. M. Deléglise and J. Rivat, *Computing $\pi(x)$: the Meissel, Lehmer, Lagarias, Miller, Odlyzko method*, Mathematics of Computation **65** (1996), no. 213, 235–245.

3. Kevin Dowd and Charles R. Severance, *High performance computing*, second ed., O'Reilly and Associates, Inc., 101 Morris Street, Sebastopol, CA 95472, 1998.
4. William F. Galway, *Robert Bennion's "hopping sieve"*, Algorithmic Number Theory (ANTS-III) (Joe P. Buhler, ed.), Lecture Notes in Computer Science, vol. 1423, Springer, Berlin, June 1998, pp. 169–178.
5. ———, *Analytic computation of the prime-counting function*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2000, (expected).
6. G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, fifth ed., Oxford University Press, Oxford, 1979.
7. M. N. Huxley, *Area, lattice points, and exponential sums*, London Mathematical Society Monographs, New Series, vol. 13, The Clarendon Press, Oxford University Press, New York, 1996, Oxford Science Publications.
8. Aleksandar Ivić, *The Riemann zeta-function*, John Wiley & Sons, New York, 1985.
9. Donald E. Knuth, *The art of computer programming*, second ed., vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, Massachusetts, 1973.
10. J. C. Lagarias, V. S. Miller, and A. M. Odlyzko, *Computing $\pi(x)$: The Meissel-Lehmer method*, Mathematics of Computation **44** (1985), no. 170, 537–560.
11. J. C. Lagarias and A. M. Odlyzko, *Computing $\pi(x)$: an analytic method*, Journal of Algorithms **8** (1987), no. 2, 173–191.
12. D. H. Lehmer and Emma Lehmer, *A new factorization technique using quadratic forms*, Mathematics of Computation **28** (1974), 625–635.
13. Thomas R. Nicely, *Enumeration to 10^{14} of the twin primes and Brun's constant*, Virginia J. Sci. **46** (1995), no. 3, 195–204.
14. Thomas R. Nicely, *New maximal prime gaps and first occurrences*, Mathematics of Computation **68** (1999), no. 227, 1311–1315.
15. Hans Riesel, *Prime numbers and computer methods for factorization*, second ed., Progress in Mathematics, vol. 126, Birkhäuser, Boston, MA, 1994.
16. Waclaw Sierpiński, *Sur un problème du calcul des fonctions asymptotiques*, 1906, Oeuvres choisies, Tome I, PWN—Éditions Scientifiques de Pologne, Warsaw, 1974, pp. 73–108.
17. Jonathan P. Sorenson, *Trading time for space in prime number sieves*, Algorithmic Number Theory (ANTS-III) (Joe P. Buhler, ed.), Lecture Notes in Computer Science, vol. 1423, Springer, Berlin, June 1998, pp. 179–195.
18. Gérald Tenenbaum, *Introduction to analytic and probabilistic number theory*, Cambridge University Press, Cambridge, 1995.
19. J. G. van der Corput, *Über Gitterpunkte in der Ebene*, Mathematische Annalen **81** (1920), 1–20.
20. Georges Voronoi, *Sur un problème du calcul des fonctions asymptotiques*, J. Reine Angew. Math. **126** (1903), 241–282.

Counting Points on Hyperelliptic Curves over Finite Fields

Pierrick Gaudry¹ and Robert Harley²

¹ LIX, École Polytechnique
91128 Palaiseau Cedex, France

² Projet Cristal, INRIA, Domaine de Voluceau - Rocquencourt
78153 Le Chesnay, France

Abstract. We describe some algorithms for computing the cardinality of hyperelliptic curves and their Jacobians over finite fields. They include several methods for obtaining the result modulo small primes and prime powers, in particular an algorithm *à la* Schoof for genus 2 using Cantor's division polynomials. These are combined with a birthday paradox algorithm to calculate the cardinality. Our methods are practical and we give actual results computed using our current implementation. The Jacobian groups we handle are larger than those previously reported in the literature.

Introduction

In recent years there has been a surge of interest in algorithmic aspects of curves. When presented with any curve, a natural task is to compute the number of points on it with coordinates in some finite field. When the finite field is large this is generally difficult to do.

René Schoof gave a polynomial time algorithm for counting points on elliptic curves i.e., those of genus 1, in his ground-breaking paper [Sch85]. Subsequent improvements by Elkies and Atkin ([Sch95], [Mor95], [Elk98]) lowered the exponent to the point where efficient implementations became possible. After further improvements ([Cou96], [Ler97]) several implementations of the Schoof-Elkies-Atkin algorithm were actually written and very large finite fields can now be handled in practice ([Mor95], [Ver99]).

For higher genus, significant theoretical progress was made by Pila who gave a polynomial time algorithm in [Pil90] (see also [HI98]). However to date these methods have not been developed as extensively as the elliptic case. As a first step towards closing this gap it is fruitful to concentrate on low genus hyperelliptic curves, as these are a natural first generalization of elliptic curves and techniques used in the elliptic case can be adapted. Such techniques include Schoof-like methods and several others which all contribute to a practical algorithm.

We mention two possible applications of the ability to count points on low genus hyperelliptic curves. An early theoretical application was the proof that primality testing is in probabilistic polynomial time, [AH92]. A practical application results from the apparent difficulty of computing discrete logarithms

in the Jacobian groups of these curves. In low genus, no sub-exponential algorithms are currently known, except for some very thin sets of examples ([Rüc99], [FR94]) and hence the Jacobian group of a random curve is likely to be suitable for constructing cryptosystems [Kob89]. To build such a cryptosystem, it is first desirable to check that the group order has a large prime factor since otherwise the logarithm could be computed in small subgroups [PH78].

We restrict ourselves to *odd* characteristic for simplicity. We will work with models of odd degree where arithmetic is analogous to that of imaginary quadratic fields. For the even degree alternative, which is similar to real quadratic fields, see the recent paper [ST99] which describes a birthday paradox algorithm optimized using an analogue of Shanks' infrastructure.

Our contribution contains several complementary approaches to the problem of finding the size of Jacobian groups, all of which have been implemented. By combining these approaches we have been able to count larger groups than previously reported in the literature.

The first approach is an efficient birthday paradox algorithm for hyperelliptic curves. We have filled in all the details required for a large-scale distributed implementation, although the basic idea has been known for 20 years. In our implementation we also use an optimized group operation for genus 2, in which we have reduced the number of field operations required.

The time taken grows as a small power of the field size and this algorithm, if used in isolation, would take a prohibitive amount of time to handle large groups such as those of cryptographic size. However our version of it can take advantage of prior information on the result modulo some integer. We elaborate various strategies for collecting as much of this information as possible.

We show that when the characteristic p is not too large, the result modulo p can be obtained surprisingly easily using the Cartier-Manin operator. It provides an elegant and self-contained method based on theoretical material proved in the 1960's.

To go further, we also extend Schoof's algorithm to genus 2 curves using Cantor's division polynomials. On the basis of previous outlines existing in the literature, but not directly implementable, we elaborated a practical algorithm and programmed it in Magma. For the case where the modulus is a power of 2, we are able to bypass computations with division polynomials and use a much faster technique based on formulae for halving in the Jacobian.

The combinations of these techniques has allowed us to count genus 2 groups with as many as 10^{38} elements.

We would particularly like to thank Éric Schost of the GAGE laboratory at École Polytechnique for helpful discussion concerning algebraic systems. Furthermore his assistance in computing Gröbner bases was invaluable and allowed us to compute group orders modulo larger powers of 2 than would otherwise have been possible.

We also thank François Morain for many constructive comments on this paper.

Prerequisites and Notations

We will take a concrete approach, concentrating on arithmetic and algorithmic aspects rather than more abstract geometric ones.

Let g be a positive integer and let \mathbb{F}_q be the finite field of $q = p^n$ elements, where p is an odd prime. For our purposes, a *hyperelliptic curve of genus g* is the set of solutions (x, y) of the equation $y^2 = f(x)$, where $f(x)$ is a monic polynomial of degree $2g+1$ with coefficients in \mathbb{F}_q and with distinct roots¹. Note that the coordinates may be in the base field \mathbb{F}_q or in an extension field.

When a point $P = (x_P, y_P)$ is on the curve \mathcal{C} , its *opposite* is the point $-P = (x_P, -y_P)$. A *divisor*² is a formal sum $D = \sum_i P_i$ of points on \mathcal{C} . Note that points may be repeated with some multiplicity in the sum. A *semi-reduced divisor* is a divisor with no two points opposite. Such a divisor with k points is said to have *weight k* . A *reduced divisor* is a semi-reduced divisor of weight $k \leq g$.

The *Jacobian*, denoted \mathbf{J} , is the set of reduced divisors. An important fact is that one can define an addition operation on reduced divisors which makes \mathbf{J} into a group, whereas this is not possible on the curve itself directly. This group law is denoted by $+$ and will be described in the next section.

A convenient representation of reduced (and semi-reduced) divisors, due to Mumford [Mum84], uses a pair of polynomials $\langle u(x), v(x) \rangle$. Here $u(x) = \prod_i (x - x_i)$ and $v(x)$ interpolates the points P_i respecting multiplicities. More precisely $v = 0$ or $\deg v < \deg u$, and u divides $f - v^2$. We say that a semi-reduced divisor is *defined over* a field \mathbb{F} when the coefficients of u and v are in \mathbb{F} (even though the coordinates x_i and y_i may be in an extension field) and write \mathbf{J}/\mathbb{F} for the set of such divisors.

Most reduced divisors have weight g . The set of those with strictly lower weight is called Θ . A divisor of weight 1 i.e., with a single point $P = (x_P, y_P)$, is represented by $\langle u(x), v(x) \rangle = \langle x - x_P, y_P \rangle$. The unique divisor of weight 0, $\mathcal{O} = \langle u(x), v(x) \rangle = \langle 1, 0 \rangle$, is the neutral element of the addition law. Scalar multiplication by an integer l is denoted by:

$$[l]D = D + D + \cdots + D \quad (l \text{ times}) \tag{1}$$

We say that D is an l -torsion divisor whenever $[l]D = \mathcal{O}$. The set of all l -torsion divisors, including those defined over extension fields, is denoted by $\mathbf{J}[l]$.

We concentrate particularly on genus-2 curves and in this case the divisors in $\mathbf{J} \setminus \Theta$ have the form:

$$D = \langle x^2 + u_1x + u_2, v_0x + v_1 \rangle . \tag{2}$$

¹ Strictly speaking, this is the affine part of a smooth projective model. In genus 2 every curve is birationally equivalent to such a curve provided the base field is large enough.

² Strictly speaking, these are degree-0 divisors with the multiplicity of the point at infinity left implicit.

1 Group Law in the Jacobian

We will sketch the group law i.e., addition of reduced divisors, using the intuitive ‘sum of points’ notation and then describe efficient formulae for computing the law in genus 2 using Mumford’s representation.

The computation of $D_1 + D_2$ can be viewed, at a high level of abstraction, as the following three steps:

- form a (non-reduced) divisor with all the points of D_1 and D_2 ,
- semi-reduce it by eliminating all pairs of opposite points,
- reduce it completely.

The third step is the only one that presents any difficulty³. When we reach it we have a semi-reduced divisor with at most $2g$ points. If there are g or fewer then no reduction is necessary but if there are more than g we reduce by a higher-genus analogue of the well known chord-and-tangent operation for elliptic curves.

1.1 Reduction Step

Fix $g = 2$ and, for the moment, consider a semi-reduced divisor R with 3 distinct points. The reduction of R is as follows.

Let $y = a(x)$ be the equation of the parabola (or perhaps line) interpolating the three points. The roots of $f - a^2$ are the abscissae of the intersections between the parabola and the curve. This is a quintic polynomial so there are five intersections (including multiplicities). We already know 3 of them, the points of R . Form a divisor S with the other two, and the result of the reduction is $-S$.

In the more frequent case where R has 4 points, choose an interpolating cubic (or lower degree) polynomial $a(x)$ instead. Then $f - a^2$ has degree 5 or 6 and we know 4 intersections. Form S with the others and the result is $-S$.

In cases where some points of R are repeated, the interpolation step is adjusted to ensure tangency to the curve with sufficient multiplicity. Also, in genus $g > 2$ the reduction step may need to be repeated several times.

In practice it would be inefficient to compute the group law this way using the representation of divisors as sums of points, since the individual points may be defined over extension fields. By using Mumford’s notation we can work entirely in the field of definition of the divisors.

1.2 Group Law in Mumford’s Notation

Cantor gave two forms of the group law using Mumford’s notation in [Can87]. One was a direct analogue of Gauss’s reduction of binary quadratic forms of

³ In a more classical treatment the reduction would be described as choosing a representative for an equivalence class of degree 0 divisors modulo linear equivalence, where linearly equivalent divisors are those that differ by a principal divisor.

negative discriminant, the other an asymptotically fast algorithm for high genus making clever use of fast polynomial arithmetic.

We describe an efficient algorithm, carefully optimized to reduce the number of operations required. We find that in genus 2 doubling a divisor or adding two divisors both take 30 multiplication operations and 2 inversions, in general. Note for comparison that optimized elliptic curve operations typically take 3 or 4 multiplications and 1 inversion.

Space limits us to a brief description of the genus 2 doubling operation. Let $D = \langle u, v \rangle$. We cover the cases that may occur, in order of increasing complexity.

Simple case: If $v = 0$ the result is simply \mathcal{O} .

Weight 1: Here $u(x) = x - x_P$ and $v(x) = y_P$. The result is $\langle (x - x_P)^2, ax + b \rangle$, where $ax + b$ is the tangent line at P with $a = f'(x_P)/2y_P$ and $b = y_P - ax_P$.

Weight 2: Compute the resultant $r = u_2v_0^2 + v_1^2 - u_1v_0v_1$ of u and v .

Resultant 0: If $r = 0$ then u and v have a root in common i.e., D has a point with ordinate 0. Isolate the other point by $x_P = -u_1 + v_1/v_0$, $y_P = v_1 + v_0x_P$ and return to the weight 1 case above.

General case: Consider the fact that v is a square root of f modulo u . We can double the multiplicity of all points in D by using a Newton iteration to compute a square root modulo u^2 .

- Newton iteration: set $U = u^2$, and $V = (v + f/v)/2 \bmod U$,
- Get ‘other’ roots: set $U = (f - V^2)/U$,
- Make U monic,
- Reduce V modulo U ,

The result is $\langle U, -V \rangle$.

Several observations help to optimize calculation with these formulae: after the first step, $V \equiv v \bmod u$; also the division by U in the second step is exact; not all coefficients of the polynomials are really needed; finally some multiplications can be avoided using Karatsuba’s algorithm.

The general addition operation is similar to doubling although the Newton iteration is replaced by a little Chinese Remainder calculation and more cases need to be handled. Since the details are somewhat tedious, we give the resulting pseudo-code and sample C code at the following Web site:

<http://cristal.inria.fr/~harley/hyper/>

2 Frobenius Endomorphism

In this section we collect some useful results and quote them without proof. A starting point for the reader interested in pursuing this material is [IR82] and the references therein.

We first describe properties of the q -power Frobenius endomorphism $\phi(x) = x^q$. Note that it has no effect on elements of \mathbb{F}_q but it becomes non-trivial in

extension fields. This map extends naturally to points, by transforming their x and y coordinates. It extends further to divisors by acting point-wise.

Crucially, this latter action is equivalent to acting on each coefficient of the u and v polynomials in Mumford's notation. When a divisor is defined over \mathbb{F}_q , ϕ may permute its points but it leaves the divisor as a whole invariant.

2.1 Characteristic Polynomial

The ϕ operator acts linearly and has a characteristic polynomial of degree $2g$ with integer coefficients. In genus 2 it is known to have the form:

$$\chi(t) = t^4 - s_1 t^3 + s_2 t^2 - s_1 q t + q^2 , \quad (3)$$

so that $\chi(\phi)$ is the identity map on all of \mathbf{J} , in other words:

$$\forall P \in \mathbf{J}, \quad \phi^4(P) - [s_1]\phi^3(P) + [s_2]\phi^2(P) - [s_1q]\phi(P) + [q^2]P = \mathcal{O} . \quad (4)$$

The so-called *Riemann hypothesis for curves*, on the roots of their zeta functions, was proved by Weil and implies that the complex roots of χ have absolute value \sqrt{q} . Hence, in genus 2 the following bounds apply: $|s_1| \leq 4\sqrt{q}$ and $|s_2| \leq 6q$.

2.2 Relations Between Frobenius and Cardinalities

The Frobenius is intimately related to the number of points on the curve and the number of divisors in \mathbf{J} , over the base field and its extensions.

First of all, knowledge of χ is equivalent to that of $\#\mathcal{C}/\mathbb{F}_{q^i}$ for $1 \leq i \leq g$. In genus 2 the following formulae relate them:

$$\#\mathcal{C}/\mathbb{F}_q = q - s_1 \quad \text{and} \quad \#\mathcal{C}/\mathbb{F}_{q^2} = q^2 - s_1^2 + 2s_2 . \quad (5)$$

Furthermore $\#\mathbf{J}/\mathbb{F}_q$ is completely determined by χ according to the formula $\#\mathbf{J}/\mathbb{F}_q = \chi(1)$. An important consequence is that the group order is constrained to a rather small interval, the Hasse-Weil interval:

$$\lceil (\sqrt{q} - 1)^{2g} \rceil \leq \#\mathbf{J}/\mathbb{F}_q \leq \lfloor (\sqrt{q} + 1)^{2g} \rfloor . \quad (6)$$

In the reverse direction, knowledge of $\#\mathbf{J}/\mathbb{F}_q$ almost determines χ for q large enough. For instance in genus 2, $(\#\mathbf{J}/\mathbb{F}_q) - q^2 - 1 = s_2 - s_1(q+1)$ and the bound on s_2 given above ensures that there are $O(1)$ possibilities.

3 Birthday Paradox Algorithm

To compute the group order $N = \#\mathbf{J}/\mathbb{F}_q$ exactly we search for it in the Hasse-Weil interval which has width w close to $4gq^{g-1/2}$. The first few coefficients s_i of χ can be computed by exhaustively counting points on the curve over \mathbb{F}_{q^i} . Doing so for $i \leq I$ reduces the search interval to width $w = O(q^{g-(I+1)/2})$ but costs $O(q^I)$ (see [Elk98]). In genus 2 this is not useful and one simply takes $w = 2\lceil 4(q+1)\sqrt{q} \rceil$.

3.1 Computing the Order of the Group

Assume for the moment that we know how to compute the order n of a randomly chosen divisor D in \mathbf{J}/\mathbb{F}_q (from now on the term ‘‘divisor’’ always refers to a reduced divisor). Writing e for the group exponent, we have $n \mid e$ and $e \mid N$ and thus N is restricted to at most $\lceil (w+1)/n \rceil$ possibilities. Usually $n \geq w$ and so N is completely determined.

It is possible for n to be smaller, though. In such a case we could try several other randomly chosen divisors, taking n to be the least common multiple of their orders and stopping if $n > w$. After a few tries n will converge to e and if $e > w$ the method terminates.

However in rare cases the exponent itself may be small, $e \leq w$. It is known that \mathbf{J}/\mathbb{F}_q is the product of at most $2g$ cyclic groups and thus $e \geq \sqrt{q} - 1$ and in fact this lower bound can be attained.

It is possible to obtain further information by determining the orders of divisors in the Jacobian group of the quadratic twist curve, but even this may not be sufficient. We do not yet have a completely satisfactory solution for such a rare case, however we mention that the Weil pairing may provide one.

3.2 Computing the Order of One Divisor

To determine the order n of an arbitrary divisor D we find some multiple of n , factor it and search for its smallest factor d such that $[d]D = \mathcal{O}$.

There are certainly multiples of n in the search interval (since the group order is one such) and we can find one of them using a birthday paradox algorithm, in particular a distributed version of Pollard’s lambda method [Pol78] with distinguished points. For a similar Pollard rho method see [vOW99].

Since the width of the search interval is w , we expect to determine the multiple after $O(\sqrt{w})$ operations in the Jacobian. By using distinguished points and distributing the computation on M machines, this takes negligible space and $O(\sqrt{w}(\log q)^2/M)$ time⁴.

The birthday paradox algorithm is as follows.

- Choose some distinguishing characteristic.
- Choose a hash function h that hashes divisors to the range 0..19, say.
- Pick 20 random step lengths $l_i > 0$ with average roughly $M\sqrt{w}$,
- Precompute the 20 divisors $D_i = [l_i]D$.
- Precompute $E = [c]D$.

Here c is the center of the search interval. In genus 2 $c = q^2 + 6q + 1$. The calculation then consists of many ‘chains’ of iterations run on M client machines:

- Pick a random $r < w$ and compute $R = [r]D$.
- Pick a random bit b and if it is 1 set R to $R + E$.
- While R is not distinguished, set $r := r + l_{h(R)}$ and $R := R + D_{h(R)}$.
- Store the distinguished R on a central server along with r and b .

⁴ Using classical algorithms for field arithmetic.

The distinguishing feature must be chosen to occur with probability significantly less than \sqrt{w}/M , say 50 times less. Thus each chain takes about $\sqrt{w}/M/50$ steps and has length about $w/50$.

Note that chains with $b = 0$ visit many pseudo-random divisors in the set $S_1 = \{[r]D \mid 0 \leq r < w\}$ and a few with larger r . Chains with $b = 1$ visit many divisors in $S_2 = \{E + [r]D \mid 0 \leq r < w\}$ and a few with larger r . However the choice of E guarantees that the intersection $I = S_1 \cap S_2$ contains at least $w/2$ divisors.

Now after a total of $O(\sqrt{w})$ steps have been performed, $O(\sqrt{w})$ divisors have been visited and $O(\sqrt{w})$ of them are expected to be in I . Then the birthday paradox guarantees a significant chance that a *useful collision* occurs i.e., that the same divisor R is visited twice with different bits b . Shortly afterwards a useful collision of distinguished points is detected at the server, between R_0 and R_1 say.

Therefore $r_0 \equiv c + r_1$ modulo n and finally $c + r_1 - r_0$ is the desired multiple of n .

3.3 Beyond the Birthday Paradox

To handle larger examples than is possible with the birthday paradox algorithm alone, we precompute the Jacobian order modulo some integer. If N is known modulo m then the search for a multiple of a divisor's order can be restricted to an arithmetic progression modulo m , rather than the entire search interval⁵. In this way the expected number of operations can be reduced by a factor \sqrt{m} .

The algorithm outlined above needs to be modified as follows (we can assume that m is much smaller than w since otherwise no birthday paradox algorithm would be required!).

- Increase the frequency of the distinguishing characteristic by a factor \sqrt{m} .
- The step lengths must be multiples of m chosen with average length $M\sqrt{wm}$.
- Replace E with $[z]D$ where z is nearest c such that $z \equiv N \pmod{m}$.

To compute N modulo m with m as large as possible, we will first compute it modulo small primes and prime powers using various techniques explained in the next few sections. Then the Chinese Remainder Theorem gives N modulo their product.

To date this use of local information has speeded up the birthday paradox algorithm by a significant factor in practice. It should be pointed out however that while the birthday paradox algorithm takes exponential time, the Schoof-like algorithm described below takes polynomial time. Hence it can be expected that for future calculations with very large Jacobians, the Schoof part will provide most of the information.

⁵ Note that we could also take advantage of partial information that restricted N to several arithmetic progressions modulo m .

4 Cartier-Manin Operator and Hasse-Witt Matrix

We propose a method for calculating the order of the Jacobian modulo the characteristic p of the base field, by using the so-called *Cartier-Manin operator* and its concrete representation as the *Hasse-Witt matrix* (see [Car57]). In the case of hyperelliptic curves, this $g \times g$ matrix can be computed by a method given in [Yui78] which generalizes the computation of the Hasse invariant for elliptic curves. Yui's result is as follows:

Theorem 1. *Let $y^2 = f(x)$ with $\deg f = 2g+1$ be the equation of a genus g hyperelliptic curve. Denote by c_i the coefficient of x^i in the polynomial $f(x)^{(p-1)/2}$. Then the Hasse-Witt matrix is given by*

$$A = (c_{ip-j})_{1 \leq i, j \leq g} . \quad (7)$$

In [Man65], Manin relates it to the characteristic polynomial of the Frobenius modulo p . For a matrix $A = (a_{ij})$, let $A^{(p)}$ denote the elementwise p -th power i.e., (a_{ij}^p) . Then Manin proved the following result:

Theorem 2. *Let C be a curve of genus g defined over a finite field \mathbb{F}_{p^n} . Let A be the Hasse-Witt matrix of C , and let $A_\phi = AA^{(p)} \cdots A^{(p^{n-1})}$. Let $\kappa(t)$ be the characteristic polynomial of the matrix A_ϕ , and $\chi(t)$ the characteristic polynomial of the Frobenius endomorphism. Then*

$$\chi(t) \equiv (-1)^g t^g \kappa(t) \mod p . \quad (8)$$

Now it is straightforward to compute $\chi(t)$ modulo the characteristic p and hence $\#\mathbf{J}/\mathbb{F}_q \mod p$, provided that p is not too large (say at most 100000). Note that this is a very efficient way to get information on the Jacobian order, particularly when p is moderately large. Such a situation can occur in practice with fields chosen, for implementation reasons, to be of the form \mathbb{F}_{p^n} with p close to a power of 2 such as $p = 2^8 - 5$ or $p = 2^{16} - 15$.

5 Algorithm à la Schoof

In this section we describe a polynomial time algorithm à la Schoof for computing the cardinality of \mathbf{J}/\mathbb{F}_q in genus 2. This algorithm follows theoretical work of Pila [Pil90] and Kampkötter [Kam91]. We make extensive use of the division polynomials described by Cantor [Can94].

5.1 Hyperelliptic Analogue of Schoof's Algorithm

The hyperelliptic analogue of Schoof's algorithm consists of computing χ modulo some small primes l by working in $\mathbf{J}[l]$. Once this has been done, modulo sufficiently many primes (or prime powers), then χ can be recovered exactly by the Chinese Remainder Theorem. From the bounds on s_i above, it suffices to

consider $l = O(\log q)$. In practice we use a few small l , determine χ modulo their product, and use this information to optimize a birthday paradox search as described previously.

Let l be a prime power co-prime with the characteristic. Then the subgroup of l -torsion points has the structure $\mathbf{J}[l] \cong (\mathbb{Z}/l\mathbb{Z})^{2g}$. Moreover, the Frobenius acts linearly on this subgroup and Tate's theorem [Tat66] states that the characteristic polynomial of the induced endomorphism is precisely the characteristic polynomial of the Frobenius endomorphism on \mathbf{J} with its coefficients reduced modulo l . Hence by computing the elements of $\mathbf{J}[l]$ and the Frobenius action on them, we can get the characteristic polynomial modulo l .

The following lemma due to Kampkötter simplifies the problem.

Lemma 1. *If l is an odd prime power, then the set $\mathbf{J} \setminus \Theta$ contains a $\mathbb{Z}/l\mathbb{Z}$ -basis of $\mathbf{J}[l]$.*

Thus the Frobenius endomorphism on $\mathbf{J}[l]$ is completely determined by its action on $\mathbf{J}[l] \setminus \Theta$.

Let $D = \langle x^2 + u_1x + u_2, v_0x + v_1 \rangle$ be a divisor in $\mathbf{J} \setminus \Theta$, then the condition $[l]D = \mathcal{O}$ can be expressed by a finite set of rational equations in u_1, u_2, v_0, v_1 . More precisely, there exists an ideal I_l of the polynomial ring $\mathbb{F}_q[U_1, U_2, V_0, V_1]$ such that D lies in $\mathbf{J}[l] \setminus \Theta$ if and only if $f(u_1, u_2, v_0, v_1) = 0$ for all polynomials f in (a generating set of) the ideal I_l . In [Kam91], Kampkötter gives explicit formulae for multivariate polynomials generating I_l .

From now on, we can represent a generic element of $\mathbf{J}[l] \setminus \Theta$ by the quotient ring $\mathbb{F}_q[U_1, U_2, V_0, V_1]/I_l$. The Frobenius action can be computed for this element and it is possible to find its minimal polynomial by brute force. The characteristic polynomial is then easy to recover (at least in the case where l is a prime) and we are done. This method due to Pila and Kampkötter has polynomial-time complexity, however it involves arithmetic on ideals which requires time-consuming computations of Gröbner bases. In the following we propose another method which avoids the use of ideals.

5.2 Cantor's Division Polynomials

In [Can94], Cantor defined *division polynomials* of hyperelliptic curves, generalizing the elliptic case, and gave an efficient recursion to build them.

These polynomials are closely related to Kampkötter's ideal I_l , but they allow a Schoof-like algorithm to work mostly with one instead of four variables. An approximate interpretation of the phenomenon is that the division polynomials lead to a representation of I_l directly computed in a convenient form (almost a Gröbner basis for a lexicographical order).

Cantor's construction provides 6 sequences of polynomials $d_0^{(l)}, d_1^{(l)}, d_2^{(l)}$ and $e_0^{(l)}, e_1^{(l)}, e_2^{(l)}$ such that for divisors $P = \langle x - x_P, y_P \rangle$ of weight 1 in general position, we get

$$[l]P = \left\langle x^2 + \frac{d_1^{(l)}(x_P)}{d_0^{(l)}(x_P)}x + \frac{d_2^{(l)}(x_P)}{d_0^{(l)}(x_P)}, y_P \left(\frac{e_1^{(l)}(x_P)}{e_0^{(l)}(x_P)}x + \frac{e_2^{(l)}(x_P)}{e_0^{(l)}(x_P)} \right) \right\rangle . \quad (9)$$

The degrees of these division polynomials are

d_0	d_1	d_2	e_0	e_1	e_2
$2l^2 - 1$	$2l^2 - 2$	$2l^2 - 3$	$3l^2 - 2$	$3l^2 - 2$	$3l^2 - 3$

By lemma 1 it is sufficient to consider divisors $D \notin \Theta$. In order to multiply $D = \langle u(x), v(x) \rangle$ by l we express it as a sum of two divisors of weight 1 i.e., we write $D = P_1 + P_2$. These divisors are given by $P_1 = \langle x - x_1, y_1 \rangle$ and $P_2 = \langle x - x_2, y_2 \rangle$ where x_1 and x_2 are the roots of $u(x)$ and $y_i = v(x_i)$. Clearly $[l]D = [l]P_1 + [l]P_2$.

The divisor D is an l -torsion divisor if and only if $[l]P_1$ and $[l]P_2$ are opposite divisors. This last condition is converted into a condition on the polynomial representations $[l]P_1 = \langle u_{P_1}(x), v_{P_1}(x) \rangle$ and $[l]P_2 = \langle u_{P_2}(x), v_{P_2}(x) \rangle$. Indeed two divisors are opposite if their u polynomials are equal and their v polynomials are opposite. Hence the elements of $\mathbf{J}[l] \setminus \Theta$ are characterized by a set of rational equations in the 4 indeterminates x_1, x_2, y_1, y_2 , two of them involving only the two indeterminates x_1 and x_2 .

Thus we get an ideal similar to I_l represented in a convenient form: we can eliminate x_2 with the two bivariate equations by computing some resultants, then we have a univariate polynomial in x_1 and for each root x_1 it is not difficult to recover the corresponding values of x_2, y_1 and y_2 .

5.3 Details of the Algorithm

Next we explain the computation of the characteristic polynomial modulo a fixed prime power l . Here we will assume that l is odd (the even case discussed in the next section).

Building an Elimination Polynomial for x_1 . We first compute Cantor's l -division polynomials. We refer to the original paper [Can94] for the recursion formulae and the proof of the construction. This phase takes negligible time compared to what follows.

The second step is to eliminate x_2 in the two bivariate equations. The system looks like

$$\begin{cases} E_1(x_1, x_2) = d_1(x_1)d_2(x_2) - d_1(x_2)d_2(x_1) = 0 \\ E_2(x_1, x_2) = d_0(x_1)d_2(x_2) - d_0(x_2)d_2(x_1) = 0 \end{cases}, \quad (10)$$

The polynomial $(x_1 - x_2)$ is clearly a common factor of E_1 and E_2 , and this factor is a parasite: it does not lead to a l -torsion divisor⁶. We throw away this factor and consider the new reduced system, still denoting the two equations by

⁶ If there is another common factor of E_1 and E_2 , we have to throw it away. This occurs when a non trivial l -torsion divisor is in Θ . The values for the degrees assume that we are in the generic case.

$E_1(x_1, x_2)$ and $E_2(x_1, x_2)$. Then we eliminate x_2 by computing the following resultant

$$R(x_1) = \text{Res}_{x_2}(E_1(x_1, x_2), E_2(x_1, x_2)) = 0 . \quad (11)$$

We can then note that $R(x_1)$ is divisible by some high power of $d_2(x_1)$. Indeed, if $d_2(x_1) = 0$ then the expressions E_1 and E_2 have common roots (at the roots of $d_2(x_2)$). The power of d_2 in R is $\delta = 2l^2 - 2$. We assume that the base field is large enough and we specialize the system at many distinct values for x_1 . Substituting ξ_i for x_1 , the system becomes two *univariate* polynomials in x_2 , for which we compute the resultant r_i . With enough pairs (ξ_i, r_i) i.e., one more than a bound on the degree of $\tilde{R}(x_1) = R(x_1)/(d_2(x_1))^\delta$, we can recover $\tilde{R}(x_1)$ by interpolation. Knowing the degrees of d_0, d_1, d_2 , it is easy to get

$$\deg \tilde{R}(x_1) = 4l^4 - 10l^2 + 6 . \quad (12)$$

Eliminating the Parasites (Optional). As previously mentioned there are l^4 divisors of l -torsion and thus the degree of $\tilde{R}(x_1)$ is too high by a factor 4. This means that there are still a lot of parasite factors, due to the fact that we only took conditions on the abscissae x_1, x_2 into account and imposed nothing on the ordinates y_1, y_2 . Two strategies can be used: we can decide to live with these parasites and go on to the next step or we can compute another resultant to eliminate them (and get a polynomial of degree $l^4 - 1$). The choice depends on the relative speeds of the resultant computation and the root-finding algorithm.

In order to eliminate the parasites we construct a third equation $E_3(x_1, x_2)$, coming from the fact that the ordinates of $[l]P_1$ and $[l]P_2$ are opposite. We write that the coefficients are opposite,

$$\begin{cases} y_1 \frac{e_1(x_1)}{e_0(x_1)} = -y_2 \frac{e_1(x_2)}{e_0(x_2)} \\ y_1 \frac{e_2(x_1)}{e_0(x_1)} = -y_2 \frac{e_2(x_2)}{e_0(x_2)} \end{cases} , \quad (13)$$

and this system implies that $E_3(x_1, x_2) = e_1(x_1)e_2(x_2) - e_1(x_2)e_2(x_1) = 0$.

Taking the resultant between E_1 and E_3 , we get a polynomial $\tilde{S}(x_1)$ of degree $12l^4 - 30l^2 + 18$ whose GCD with $\tilde{R}(x_1)$ is of degree $l^4 - 1$ (in general, a few parasites may remain in rare cases). We still denote this GCD by $\tilde{R}(x_1)$ for convenience.

Recovering the Result Modulo l . To find the result we factor $\tilde{R}(x_1)$ and, for each irreducible factor, we construct an extension of \mathbb{F}_q using this factor to get a root X_1 of $\tilde{R}(x_1)$. Then we substitute this root into E_1 and E_2 and recover the corresponding root X_2 . Using the equation of the curve we get the ordinates Y_1 and Y_2 , which may be in a quadratic extension. We get the two divisors $P_1 = \langle x - X_1, Y_1 \rangle$ and $P_2 = \langle x - X_2, Y_2 \rangle$ and check whether $[l](P_1 + P_2) = \mathcal{O}$ or $[l](P_1 - P_2) = \mathcal{O}$. If neither holds, then we started from a parasite solution and

we try another factor of $\tilde{R}(x_1)$. In the favorable case we get an l -torsion divisor D with which we check the Frobenius equation. To do so we compute

$$[s_1]\phi^3(D) + [qs_1 \bmod l]\phi(D) , \quad (14)$$

for every $s_1 \in [0, l - 1]$ and

$$\phi^4(D) + [s_2]\phi^2(D) + [q^2 \bmod l]D , \quad (15)$$

for every $s_2 \in [0, l - 1]$. We only keep the pairs (s_1, s_2) for which these are equal.

If there is only one pair (s_1, s_2) left, or if there are several pairs all leading to the same value for the cardinality modulo l , then it is not necessary to continue with another factor. Thus it is usually not necessary to have a complete factorization of $\tilde{R}(x_1)$ and the computation is faster if one starts with irreducible factors of smallest degree.

We summarize the above in the following:

Algorithm. Computation of $\#\mathbf{J}/\mathbb{F}_q$ modulo l .

1. Compute $\tilde{R}(x_1)$.
2. Find a factor of $\tilde{R}(x_1)$ of smallest degree.
3. Build P_1 and P_2 with this factor.
4. Check if $P_1 + P_2$ or $P_1 - P_2$ is an l -torsion divisor. If so call it D , else go back to step 2.
5. For each remaining pair (s_1, s_2) , check the Frobenius equation for D .
6. Compute the set of possible values of $\#\mathbf{J}/\mathbb{F}_q$ from the remaining values of (s_1, s_2) . If there are several values left, go back to step 2. If there is just one, return it.

5.4 Complexity

We evaluate the cost of this algorithm by counting the number of operations in the base field \mathbb{F}_q . We neglect all the $\log^\alpha l$ factors, and denote by $M(x)$ the number of field operations required to multiply two polynomials of degree x .

The first step requires $O(l^4)$ resultant computations, each of which can be done in $M(l^2)$ operations, and the interpolation of a degree $O(l^4)$ polynomial which can be done in $M(l^4)$ operations. For the analysis of the remaining steps, we will denote by d the degree of the smallest factor of $\tilde{R}(x_1)$ that allows us to conclude. We assume moreover that the most costly part of the factorization is the distinct degree factorization (which is the case if d is small and if the number of factors of degree d is not too large). Then the cost of finding the factor is $O(d \log(q))M(l^4)$. Thereafter the computation relies on manipulations of polynomials of degree d and the complexity is $O(l + \log(q))M(d)$, where l reflects the l possible values of s_1 and of s_2 and $\log(q)$ reflects the Frobenius computations. Hence the (heuristic) overall cost for the algorithm is

$$O(l^4)M(l^2) + O(d \log q)M(l^4) + O(l^2 + \log q)M(d) \quad (16)$$

operations in the base field.

Now we would like to obtain a complexity for the whole Schoof-like algorithm. For that we will keep only the primes l for which $d = O(l)$; this should occur heuristically with a fixed probability (this is an analogue of ‘Elkies primes’ for elliptic curves). Then we have to use a set of $O(\log q)$ primes l , each of them satisfying $l = O(\log q)$. Moreover we will assume fast polynomial arithmetic and thus $M(x) = O(x)$ (ignoring logarithmic factors). Hence the cost of the algorithm is heuristically $O(\log^7 q)$ operations in \mathbb{F}_q . Each operation can be performed in $O(\log^2 q)$ bit operations using classical arithmetic and we get that the complexity of the Schoof-like algorithm is $O(\log^9 q)$.

Remark. This analysis is heuristic, but one could obtain a rigorous proof that the algorithm runs in polynomial time. The algorithm could also be made deterministic by avoiding polynomial factorizations. However in both cases the exponent would be higher than 9.

6 Lifting the 2-Power Torsion Divisors

In this section, we will show how to obtain some information on the $\#\mathbf{J}/\mathbb{F}_q$ modulo small powers of 2. Factoring f gives some information immediately. To go further we iterate a method for ‘halving’ divisors in the Jacobian. This quickly leads to divisors defined over large extensions, so that the run-time grows exponentially. In practice we can use this technique to obtain partial information modulo 256, say.

The divisors of order 1 or 2 are precisely the $D = \langle u(x), 0 \rangle$ for which $u(x)$ divides $f(x)$ and is of degree at most g . When f has n irreducible factors, then it has 2^n factors altogether. Exactly half of them have degree at most g , since f is square-free of degree $2g + 1$. Hence the number of such divisors is 2^{n-1} , and $2^{n-1} \mid \#\mathbf{J}/\mathbb{F}_q$. Furthermore, when f is irreducible then the 2-part is trivial and $\#\mathbf{J}/\mathbb{F}_q$ is odd.

6.1 Halving in the Jacobian

Let $D = \langle u(x), v(x) \rangle$ be a divisor different from \mathcal{O} . We would like to find a divisor Δ such that $[2]\Delta = D$. Note that there are 2^{2g} solutions, any two of which differ by a 2-torsion divisor. In general, Δ is defined over an extension of the field of definition of D .

Writing $\Delta = \langle \tilde{u}(x), \tilde{v}(x) \rangle$, we derive a rational expression for the divisor $[2]\Delta$ using the formulae of section 1. Then equating this expression with D , we get a set of $2g$ polynomial equations in $2g$ indeterminates \tilde{u}_i and \tilde{v}_i with $2g$ parameters u_i and v_i . There are g^2 such systems corresponding to the different possible weights of D and Δ .

We consider the most frequent case where D and Δ are both of weight g . The corresponding system has at most 2^{2g} solutions and these can be obtained by constructing a Gröbner basis for a lexicographical order, factoring the last

polynomial in the basis and propagating the solution to the other polynomials. All this can be done in time polynomial in $\log q$ provided that the divisor D we are dealing with is defined over an extension of bounded degree of \mathbb{F}_q .

In order to speed up the computations in the case where D is defined over a large extension, we can avoid repeated Gröbner-basis computations and instead compute a single *generic* Gröbner basis for the system, where the coefficients of D are parameters. As the halving is algebraic over \mathbb{F}_q (because the curve is defined over \mathbb{F}_q), the generic basis is also defined over \mathbb{F}_q . After this computation we can halve any divisor D , even when defined over a large extension, by plugging its coefficients into the generic basis to get the specialized one.

We are indebted to Eric Schost who kindly performed the construction of this generic Gröbner basis for the curves we studied [Sch]. For his construction, he made use of the **Kronecker** package [Lec99] written by Grégoire Lecerf. This package behaves very well on these types of problem (lifting from specialized systems to generic ones), and it is likely that we would not have been able to do this lifting by using classical algorithms for Gröbner-basis computations.

Example. Let \mathcal{C} be defined by

$$y^2 = x^5 + 1597x^4 + 1041x^3 + 5503x^2 + 6101x + 1887 , \quad (17)$$

over the finite field \mathbb{F}_p with $p = 10^{17} + 3$. We will search for all *rational* 2-power torsion divisors i.e., those defined over \mathbb{F}_p . Two irreducible factors of $f(x)$ have degree at most 2, they are

$$f_1 = x + 28555025517563816 \text{ and } f_2 = x + 74658844563359755 ,$$

Thus there are three rational divisors of order two: $P_1 = \langle f_1, 0 \rangle$, $P_2 = \langle f_2, 0 \rangle$ and $P_1 + P_2$. The halving method applied to P_1 finds four rational divisors of order 4. They are $\langle u, v \rangle$ and $\langle u, -v \rangle$ where:

$$\begin{aligned} u &= x^2 + 1571353025997967x + 12198441063534328 \\ v &= 32227723250469108x + 68133247565452990 \end{aligned}$$

and:

$$\begin{aligned} u &= x^2 + 70887725815800572x + 94321182398888258 \\ v &= 42016761890161508x + 3182371156137467 . \end{aligned}$$

There are 16 solutions altogether but the others are in extension fields (the Gröbner bases are too large to include them here!) Applying the method to P_2 and to $P_1 + P_2$ finds no further rational 4-torsion divisors. By continuing in the same manner one finds 8 divisors of order 8, 16 of order 16, 32 of order 32 and no more. Thus the 2-part of the rational Jacobian is of the form $(\mathbb{Z}/2) \times (\mathbb{Z}/32)$ and hence $\#\mathbf{J}/\mathbb{F}_p \equiv 64 \pmod{128}$.

This type of exhaustive search in the base field determines the exact power of 2 dividing $\#\mathbf{J}/\mathbb{F}_p$. In the next section we show how to find information modulo larger powers of 2.

6.2 Algorithm for Computing $\#\mathbf{J}/\mathbb{F}_q \bmod 2^k$

Next we go into extension fields to find some 2^k -torsion divisors and we substitute them into χ , the characteristic equation of the Frobenius endomorphism, to determine values of its coefficients modulo 2^k and hence the value of $\#\mathbf{J}/\mathbb{F}_q \bmod 2^k$, for increasing k .

Algorithm (for $g = 2$).

1. Factor f to find a 2-torsion divisor. Halve it to get a 4-torsion divisor D_4 .
2. Find the pair $(s_1, s_2) \bmod 4$ for which $\chi(D_4) = \mathcal{O}$. Set k to 2.
3. Compute the generic Gröbner basis for halving (weight 2) divisors in the given Jacobian.
4. Build a 2^{k+1} -torsion divisor $D_{2^{k+1}}$ by substituting the coefficients of D_{2^k} in the system, computing a root of the eliminating polynomial in an extension of minimal degree, and propagating it throughout the system.
5. For each pair $(s_1, s_2) \bmod 2^{k+1}$ compatible with the previously found pair modulo 2^k , plug $D_{2^{k+1}}$ into χ and find the pair for which $\chi(D_{2^{k+1}}) = \mathcal{O}$.
6. Set $k = k + 1$, and go back to Step 4.

Note that this is an idealized description of the algorithm. In fact there will frequently be several pairs (s_1, s_2) remaining after checking the Frobenius equation for one 2^k -torsion divisor. We can eliminate false candidates by checking with other 2^k -torsion divisors. It can be costly to eliminate all of them when the required divisors are in large extensions; an alternative strategy is to continue and expect the false candidates to be eliminated later using 2^{k+1} -torsion divisors.

In this algorithm, we could skip step 3 and compute specific Gröbner bases at each time in step 4. However, the generic Gröbner basis is more efficient and allows one to perform one or two extra iterations for the same run-time.

7 Combining these Algorithms — Practical Results

We have implemented all these algorithms and tested their performance for real computation. Some of them were written in the C programming language, and others were implemented in the Magma computer algebra system [BC97].

7.1 Prime Field

In the case where the curve is defined over a prime field \mathbb{F}_p , where p is a large prime, we use all the methods described in previous sections except for Cartier-Manin. We give some data for a ‘random’ curve for which we computed the cardinality of the Jacobian. Let the curve \mathcal{C} be defined by

$$\begin{aligned} y^2 = & x^5 + 3141592653589793238 x^4 + 4626433832795028841 x^3 \\ & + 9716939937510582097 x^2 + 4944592307816406286 x \\ & + 2089986280348253421 , \end{aligned} \tag{18}$$

over the prime field of order $p = 10^{19} + 51$. The cardinality of its Jacobian is

$$\#\mathbf{J} = 9999999982871020671452277000281660080 , \quad (19)$$

and the characteristic polynomial of the Frobenius has coefficients:

$$s_1 = 1712898036 \text{ and } s_2 = 11452277089352355350 .$$

The first step of this computation is to factor $f(x)$. It has 3 irreducible factors, thus we already know that $\#\mathbf{J} \equiv 0 \pmod{4}$.

The second step is to lift the 2-power torsion divisors. The computation of the generic halving Gröbner basis (done by E. Schost) took about one hour on an Alpha workstation. Then we lifted the divisors several times and checked the Frobenius equation. In the following table we give the degree of the extension where we found a 2^k -torsion divisor, and the information on $\#\mathbf{J}$ that we got (timings on a Pentium 450).

$\#\mathbf{J}$	deg of ext	$\#\mathbf{J}$	deg of ext	time
0 mod 2	1	16 mod 32	16	
0 mod 4	1	48 mod 64	32	
0 mod 8	4	48 mod 128	64	5000 sec
0 mod 16	8	176 mod 256	128	9 hours

The next step is to perform the Schoof-like algorithm. We did so for the primes $l \in \{3, 5, 7, 11, 13\}$. The following table gives the degree of the polynomial $\tilde{R}(x_1)$ for each l , and the smallest extension where we found an l -torsion divisor (timings on a Pentium 450).

l	degree of $\tilde{R}(x_1)$	degree of ext	$\#\mathbf{J}$	time
3	240	2	1 mod 3	1200 sec
5	2256	1	0 mod 5	300 sec
7	9120	6	4 mod 7	12 hours
11	57360	1	0 mod 11	19 hours
13	112560	7	9 mod 13	205 hours

The run-time for $l = 3$ is surprisingly large in this table. For our curve, an unlucky event occurs, which becomes rare as l increases. Indeed, after testing the Frobenius equation for *all* the 3-torsion divisors several candidates (s_1, s_2) still remain, yielding several possibilities for $\#\mathbf{J} \pmod{3}$. What this means is that the minimal polynomial of ϕ is not the characteristic polynomial. Each remaining candidate for (s_1, s_2) gives a multiple of the minimal polynomial. By taking their GCD we obtain the exact minimal polynomial, from which we can deduce the characteristic polynomial⁷ and $\#\mathbf{J} \pmod{3}$.

In our case, there are 3 pairs left after testing all the 3-torsion points, leading to the following candidates for $\#\mathbf{J} \pmod{3}$.

$(s_1, s_2) \pmod{3}$	$\#\mathbf{J} \pmod{3}$	$\chi(t) \pmod{3}$
(0, 2)	1	$t^4 - t^2 + 1$
(1, 2)	2	$t^4 - t^3 - t^2 - t + 1$
(2, 2)	0	$t^4 + t^3 - t^2 + t + 1$

⁷ See [Kam91] for more about this.

The third case is impossible because if $\#\mathbf{J} \equiv 0 \pmod{3}$ then we would have found a rational 3-torsion divisor earlier. In order to decide between the two first cases we determine the minimal polynomial, which is $t^2 + 1$ and thus the characteristic polynomial must be $(t^2 + 1)^2$ and finally $\#\mathbf{J} \equiv 1 \pmod{3}$.

However to do this we have to build all the 3-torsion divisors. This explains why the running time is higher than for $l = 5$, where we found a rational 5-torsion divisor and immediately deduced that $\#\mathbf{J} = 0 \pmod{5}$.

The final step is the birthday paradox computation. The width of the Hasse-Weil interval is roughly 2.5×10^{29} . The search space is reduced by a factor $2^8 \times 3 \times 5 \times 7 \times 11 \times 13 = 3843840$ leaving 6.6×10^{22} candidates. The search was performed on ten Alpha workstations working in parallel and calculated 5×10^{11} operations in the Jacobian. On a single 500 MHz workstation, this computation would have taken close to 50 days.

7.2 Non-prime Fields

Let \mathcal{C} be a genus 2 curve defined over \mathbb{F}_{p^n} , where p is a small odd prime. We assume that \mathcal{C} is not defined over a small subfield, for in that case it is easy to compute $\chi(t)$ using a theorem due to Weil.

Here the first step is to use Cartier-Manin to get $\chi(t) \pmod{p}$ quickly and then continue as before, except that we avoid $l = p$ in the Schoof part.

Examples: We did not try to build big examples, however we give two medium ones. For the first, let the curve \mathcal{C} be defined by

$$y^2 = x^5 + x^4 + x^3 + x^2 + tx + 1 , \quad (20)$$

over the finite field $\mathbb{F}_{3^{30}} = \mathbb{F}_3[t]/(t^{30} + t - 1)$. The cardinality of its Jacobian is

$$\#\mathbf{J} = 42391156018493425614913594804 . \quad (21)$$

The second example illustrates the advantage given by Cartier-Manin in a favorable case where $p = 2^{16} - 15$. Let the curve \mathcal{C} be defined by

$$y^2 = x^5 + x^4 + x^3 + x^2 + x + t , \quad (22)$$

over the finite field $\mathbb{F}_{p^4} = \mathbb{F}_p[t]/(t^4 - 17)$. The Cartier-Manin computation gave us $\#\mathbf{J} \equiv 58976 \pmod{p}$ in 17 minutes, and finishing using our other methods gave

$$\#\mathbf{J} = 339659790214687297284652908385855015466 . \quad (23)$$

8 Perspectives for Further Research

The present paper reports on practical algorithms for counting points on hyperelliptic curves over large finite fields and on implementations for genus 2. Although it is now possible to deal with almost cryptographic-size Jacobians, there is still a substantial amount of work to be done. Some improvements or generalizations seem to be accessible in the near future, whereas others are still quite vague. Among them we would like to mention:

- Extension of the algorithm to even characteristic. This is only a matter of translating the formulae, in order to deal with an equation of the form $y^2 + h(x) y = f(x)$. The Cartier-Manin part and the lifting of the 2-torsion should merge, giving an efficient way to compute the result modulo 2^k . For the Schoof-like part, the formulae of Cantor’s division polynomials have to be adapted, which does not appear to be too difficult.
- Extension of the Schoof-like algorithm to genus $g > 2$. The main difficulty is that it does not appear possible to avoid manipulation of ideals.
- More use could certainly be made of the Jacobian of the twist curve.
- We believe that it may be possible to lift the curve to a local field with residue field \mathbb{F}_q and use Cartier-Manin to compute $\chi(t)$ modulo small powers of the characteristic. We do not yet know how to compute the lift, however.
- A major improvement would be to elaborate a genus 2 version of the Elkies-Atkin approach for elliptic curves, which would lead to computations with polynomials of lower degree. We conjecture that it is possible to work with degrees reduced from $O(l^4)$ to $O(l^3)$. The first task is to construct modular equations for Siegel modular forms, instead of classical ones. This requires a description of isogenies for each small prime degree, which can be given by lists of cosets under left actions of the symplectic group $Sp_4(\mathbb{Z})$ instead of the classical modular group $SL_2(\mathbb{Z})$. Starting points for studying the relevant forms and groups include [Fre83] and [Kli90]. This will be explained in more detail elsewhere [Har].

All the above is the subject of active research.

References

- [AH92] L. M. Adleman and M.-D. A. Huang. *Primality testing and Abelian varieties over finite fields*, vol. 1512 of *Lecture Notes in Math.* Springer-Verlag, 1992.
- [BC97] W. Bosma and J. Cannon. *Handbook of Magma functions*, 1997. Sydney, <http://www.maths.usyd.edu.au:8000/u/magma/>.
- [Can87] D. G. Cantor. Computing in the Jacobian of an hyperelliptic curve. *Math. Comp.*, 48(177):95–101, 1987.
- [Can94] D. G. Cantor. On the analogue of the division polynomials for hyperelliptic curves. *J. Reine Angew. Math.*, 447:91–145, 1994.
- [Car57] P. Cartier. Une nouvelle opération sur les formes différentielles. *C. R. Acad. Sci. Paris Sér. I Math.*, 244:426–428, 1957.
- [Cou96] J.-M. Couveignes. Computing l -isogenies using the p -torsion. In H. Cohen, editor, *Algorithmic Number Theory*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 59–65. Springer Verlag, 1996. Second International Symposium, ANTS-II, Talence, France, May 1996, Proceedings.
- [Elk98] N. Elkies. Elliptic and modular curves over finite fields and related computational issues. In D.A. Buell and J.T. Teitelbaum, editors, *Computational Perspectives on Number Theory*, pages 21–76. AMS/International Press, 1998. Proceedings of a Conference in Honor of A.O.L. Atkin.
- [FR94] G. Frey and H.-G. Rück. A remark concerning m -divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206):865–874, April 1994.

- [Fre83] E. Freitag. *Siegelsche Modulfunktionen*, volume 254 of *Grundlehren der mathematischen Wissenschaften*. Springer–Verlag, 1983.
- [Har] R. Harley. On modular equations in genus 2. In preparation.
- [HI98] M.-D. Huang and D. Ierardi. Counting points on curves over finite fields. *J. Symbolic Comput.*, 25:1–21, 1998.
- [IR82] K. F. Ireland and M. Rosen. *A classical introduction to modern number theory*, volume 84 of *Graduate texts in Mathematics*. Springer–Verlag, 1982.
- [Kam91] W. Kampkötter. *Explizite Gleichungen für Jacobische Varietäten hyperelliptischer Kurven*. PhD thesis, Univ. Gesamthochschule Essen, August 1991.
- [Kli90] H. Klingen. *Introductory lectures on Siegel modular forms*, vol. 20 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1990.
- [Kob89] N. Koblitz. Hyperelliptic cryptosystems. *J. of Cryptology*, 1:139–150, 1989.
- [Lec99] G. Lecerf. Kronecker, *Polynomial Equation System Solver, Reference manual*, 1999. <http://www.gage.polytechnique.fr/~lecerf/software/kronecker>.
- [Ler97] R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. Thèse, École polytechnique, June 1997.
- [Man65] J. I. Manin. The Hasse-Witt matrix of an algebraic curve. *Trans. Amer. Math. Soc.*, 45:245–264, 1965.
- [Mor95] F. Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques. *J. Théor. Nombres Bordeaux*, 7:255–282, 1995.
- [Mum84] D. Mumford. *Tata lectures on theta II*, volume 43 of *Progr. Math.* Birkhauser, 1984.
- [PH78] S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Trans. Inform. Theory*, IT-24:106–110, 1978.
- [Pil90] J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, 55(192):745–763, October 1990.
- [Pol78] J. M. Pollard. Monte Carlo methods for index computation mod p . *Math. Comp.*, 32(143):918–924, July 1978.
- [Rück99] H. G. Rück. On the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 68(226):805–806, 1999.
- [Sch] E. Schost. Computing parametric geometric resolutions. Submitted to ISSAC’2000.
- [Sch85] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p . *Math. Comp.*, 44:483–494, 1985.
- [Sch95] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7:219–254, 1995.
- [ST99] A. Stein and E. Teske. Catching kangaroos in function fields. Preprint, March 1999.
- [Tat66] J. Tate. Endomorphisms of Abelian varieties over finite fields. *Invent. Math.*, 2:134–144, 1966.
- [Ver99] F. Vercauteren. #EC($GF(2^{1999})$). E-mail message to the NMBRTHRY list, Oct 1999.
- [vOW99] P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *J. of Cryptology*, 12:1–28, 1999.
- [Yui78] N. Yui. On the jacobian varieties of hyperelliptic curves over fields of characteristic $p > 2$. *J. Algebra*, 52:378–410, 1978.

Modular Forms for $\mathrm{GL}(3)$ and Galois Representations

Bert van Geemen¹ and Jaap Top²

¹ Dipartimento di Matematica, Università di Pavia

Via Ferrata 1, 27100 Pavia, Italy

geemen@dragon.ian.pv.cnr.it

² IWI, Rijksuniversiteit Groningen

Postbus 800, NL-9700 AV Groningen, The Netherlands

top@math.rug.nl

Abstract. A description and an example are given of numerical experiments which look for a relation between modular forms for certain congruence subgroups of $\mathrm{SL}(3, \mathbb{Z})$ and Galois representations.

1 Introduction

In this paper we review a recently discovered relation between some modular forms for congruence subgroups of $\mathrm{SL}(3, \mathbb{Z})$ and three dimensional representations of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ (see [vG-T] and [GKTV]). This relation is the equality of local L -factors, for primes $p \leq 173$, attached to the modular forms and to the Galois representation, see Theorem 4.5. The result gives some evidence for general conjectures of Langlands and Clozel [C1].

The first three section follow closely the notes from a seminar talk of the first author at the séminaire de théorie des nombres de Paris in January 1995. In the first section we briefly recall an instance of the relation between elliptic modular forms and Galois representations. In the second section we introduce the modular forms for $\mathrm{GL}(3)$ and the Galois representations are discussed in section three.

In section four we give some new examples of non-cusp forms for congruence subgroups of $\mathrm{SL}(3, \mathbb{Z})$ and we describe many of these in terms of classical modular forms for congruence subgroups of $\mathrm{SL}(2, \mathbb{Z})$. The last section deals with a Hodge theoretical aspect of the algebraic varieties (motives in fact) we used to define the Galois representations.

It is a pleasure to thank Avner Ash, Kevin Buzzard, Bas Edixhoven and Jasper Scholten, especially for their interest and help concerning Sect. 5.5.

2 Modular Forms: The $\mathrm{GL}(2)$ Case

Let $S_2(N)$ be the space of cusp forms of weight two for the congruence subgroup $\Gamma_0(N) \subset \mathrm{SL}(2, \mathbb{Z})$. Let $f = \sum a_n e^{2\pi i n} \in S_2(N)$ be a newform, thus $a_1 = 1$ and f

is an eigenform for the Hecke algebra: $T_p f = a_p f$ for all prime numbers p which do not divide N . For such a prime p one defines the local L -factor of f as

$$L_p(f, s) := (1 - a_p p^{-s} + p^{1-2s})^{-1},$$

note that $L_p(f, s)$ is determined by the eigenvalue a_p .

In case all a_p are in \mathbb{Z} , f defines an elliptic curve E_f , defined over \mathbb{Q} (E_f is a subvariety of the Jacobian of the modular curve $X_0(N)$). The Galois group $\text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts on the ℓ^n -torsion points of this curve which gives an ℓ -adic representation:

$$\rho_{f,\ell} : \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \longrightarrow \text{GL}_2(\mathbb{Q}_\ell).$$

The local L -factor of this representation for primes p as above does not depend on the choice of the prime $\ell \neq p$ and is defined by

$$L_p(\rho_f, s) := \det(I - \rho_{f,\ell}(F_p)p^{-s})^{-1} = (1 - \text{trace}(\rho_{f,\ell}(F_p))p^{-s} + p^{1-2s})^{-1},$$

with $F_p \in \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ a Frobenius element at p .

The Eichler-Shimura congruence relation asserts that

$$a_p = \text{trace}(\rho_{f,\ell}(F_p)) \quad \text{so} \quad L_p(f, s) = L_p(\rho_f, s)$$

(again with p a prime not dividing $N\ell$). Thus we have a method to associate to a newform f a (compatible system of ℓ -adic) Galois representation(s) $\rho_{f,\ell}$ such that the L -factors agree. This construction has been generalized to newforms of any weight (and arbitrary Hecke eigenvalues) by Deligne [D] using Galois representations on certain etale cohomology groups of certain ℓ -adic sheaves on the modular curve $X_0(N)$.

It is a pleasure to observe that recently Wiles proved a partial inverse to the construction sketched above: he shows that for a certain class of elliptic curves defined over \mathbb{Q} the corresponding Galois L -series are the L -series of newforms. As is well known, this has been used to prove Fermat's Last Theorem.

3 Modular Forms for $\text{GL}(3)$

3.1

One can also define modular forms, a Hecke algebra and local L -factors for congruence subgroups of $\text{SL}(3, \mathbb{Z})$, see below. However, the upper half plane

$$\mathbb{H} = \{z \in \mathbb{C} : \Im(z) > 0\} \cong \text{SL}(2, \mathbb{R})/\text{SO}(2),$$

which has a complex structure, is now replaced by $\text{SL}(3, \mathbb{R})/\text{SO}(3)$ (see [AGG]), a real variety of dimension 5 which, for dimension reasons(!), cannot have a complex structure.

In particular, one does not know how to associate algebraic varieties to congruence subgroups of $\text{SL}(3, \mathbb{Z})$ (in contrast to the modular curves in the $\text{GL}(2)$ -case). Therefore there are no a priori given Galois representations on etale cohomology groups which could be related to modular forms for such congruence subgroups.

3.2

In the case of $\mathrm{SL}(2, \mathbb{Z})$, the space of holomorphic modular forms of weight two for a congruence subgroup Γ is a subspace of the cohomology group $H^1(\Gamma, \mathbb{C})$. This generalizes as follows.

3.3

From now on we use the following definition:

$$\Gamma_0(N) = \left\{ (a_{ij}) \in \mathrm{SL}(3, \mathbb{Z}) \mid a_{21} \equiv 0 \pmod{N} \text{ and } a_{31} \equiv 0 \pmod{N} \right\}.$$

The modular forms for $\Gamma_0(N)$ we consider are elements of $H^3(\Gamma_0(N), \mathbb{C})$. To compute this vector space, we introduce a finite set:

$$\mathbb{P}^2(\mathbb{Z}/N) = \left\{ (\bar{x}, \bar{y}, \bar{z}) \in (\mathbb{Z}/N)^3 \mid \bar{x}\mathbb{Z}/N + \bar{y}\mathbb{Z}/N + \bar{z}\mathbb{Z}/N = \mathbb{Z}/N \right\} / (\mathbb{Z}/N)^\times.$$

When the elements of this set are viewed as column vectors, there is a natural left action of $\mathrm{SL}(3, \mathbb{Z})$ on $\mathbb{P}^2(\mathbb{Z}/N)$. This action is transitive, and the stabilizer of $(\bar{1}: \bar{0}: \bar{0})$ equals $\Gamma_0(N)$. Therefore

$$\mathrm{SL}(3, \mathbb{Z})/\Gamma_0(N) \cong \mathbb{P}^2(\mathbb{Z}/N).$$

This relation between $\Gamma_0(N)$ and $\mathbb{P}^2(\mathbb{Z}/N)$ leads to a very concrete description of the vector space $H^3(\Gamma_0(N), \mathbb{C})$. In fact, its dual $H_3(\Gamma_0(N), \mathbb{C})$ can be computed as follows:

3.4 Theorem.

([AGG], Thm 3.2, Prop 3.12)

There is a canonical isomorphism between $H_3(\Gamma_0(N), \mathbb{C})$ and the vector space of mappings $f : \mathbb{P}^2(\mathbb{Z}/N) \rightarrow \mathbb{C}$ that satisfy

1. $f(\bar{x}: \bar{y}: \bar{z}) = -f(-\bar{y}: \bar{x}: \bar{z})$,
2. $f(\bar{x}: \bar{y}: \bar{z}) = f(\bar{z}: \bar{x}: \bar{y})$,
3. $f(\bar{x}: \bar{y}: \bar{z}) + f(-\bar{y}: \bar{x}: \bar{y}: \bar{z}) + f(\bar{y} - \bar{x}: -\bar{x}: \bar{y}) = 0$.

3.5

For any $\alpha \in \mathrm{GL}(3, \mathbb{Q})$ one has a (\mathbb{C} -linear) Hecke operator:

$$T_\alpha : H^3(\Gamma_0(N), \mathbb{C}) \longrightarrow H^3(\Gamma_0(N), \mathbb{C}).$$

The adjoint operator T_α^* on the dual space $H_3(\Gamma_0(N), \mathbb{C})$ can be explicitly computed using modular symbols.

The Hecke algebra \mathcal{T} is defined to be the subalgebra of $\mathrm{End}(H^3(\Gamma_0(N), \mathbb{C}))$ generated by the T_α 's with $\det(\alpha)$ relatively prime with N . The Hecke algebra

is a commutative algebra and we are interested in eigenforms $F \in H^3(\Gamma_0(N), \mathbb{C})$ for the Hecke algebra:

$$TF = \lambda(T)F, \quad \text{with } \lambda : \mathcal{T} \rightarrow \mathbb{C} \quad (\text{for all } T \in \mathcal{T}).$$

Of particular interest are the Hecke operators $E_p = T_{\alpha_p}$, which are for a prime p not dividing N defined using $\alpha_p = \begin{pmatrix} p & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \in \mathrm{GL}(3, \mathbb{Q})$.

Let $a_p := \lambda(E_p)$, for a (given) character λ of \mathcal{T} and a prime p not dividing N , then the local L -factor of a Hecke eigenform $F \in H^3(\Gamma_0(N), \mathbb{C})$ (with the additional condition that F is cuspidal) corresponding to λ (so $E_p F = a_p F$) is

$$L_p(F, s) = (1 - a_p p^{-s} + \bar{a}_p p^{1-2s} - p^{3-3s})^{-1},$$

where \bar{a}_p is the complex conjugate of a_p . The field $K_F := \mathbb{Q}(\dots, a_p, \dots)$ generated by the Hecke eigenvalues of an eigenform F is known to be either totally real or is a CM field (a degree 2, non-real extension of a totally real field).

3.6

In [GKTV], a list of the a_p 's with $p \leq 173$ is given for several eigenforms with $N \leq 245$. Here we list some a_p 's of three particularly interesting eigenforms (these eigenforms are uniquely determined by their level N and the a_p 's listed). In case p divides N we write $**$ for a_p . In the three cases listed here $K_F = \mathbb{Q}(i)$ with $i^2 = -1$. The complex conjugates of the a_p 's for a given F are the Hecke eigenvalues for another modular form G of the same level.

$p =$	2	3	5	7	11	13	17	101	173
N	eigenvalue a_p								
128	**	$1 + 2i$	$-1 - 4i$	$1 + 4i$	$-7 - 10i$	$-1 + 4i$	7	$-105 - 100i$	$-49 - 188i$
160	**	$1 + 2i$	**	$1 - 2i$	$-3 - 12i$	$-5 - 8i$	-5	$-33 + 64i$	$99 + 104i$
205	-1	$1 + 2i$	**	$1 + 2i$	$-7 - 10i$	$3 - 8i$	-5	$115 - 40i$	$-153 - 288i$

4 Galois Representations

4.1

We are interested in relating Hecke eigenforms and Galois representations. In particular, given a Hecke eigenform F we would like to find (a compatible system of) λ -adic Galois representations

$$\rho_{F,\lambda} : \mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \longrightarrow \mathrm{GL}(W_\lambda)$$

having the same local L -factors as F . Here λ is a prime in a finite extension K_λ of \mathbb{Q}_ℓ and W_λ is a (finite dimensional) K_λ vector space. The local L -factors of

$\rho_{F,\lambda}$ (again independent of λ) being defined as before (for unramified primes, conjecturally those not dividing $N\ell$):

$$L_p(\rho_F, s) := \det(I - \rho_{F,\lambda}(F_p)p^{-s})^{-1}.$$

In particular, we want $\dim W_\lambda = 3$.

4.2

The case that K_F is totally real is analyzed by Clozel [C2]. We just recall that if in this case such a Galois representation $\rho_{F,\lambda}$ exists then $\rho_{F,\lambda}$ is selfdual in the following sense.

Consider the Tate-twisted dual Galois representation:

$$\rho_{F,\lambda}^* := {}^t \rho_{F,\lambda}^{-1}(-2) : \mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \longrightarrow \mathrm{GL}(W_\lambda), \quad \text{so} \quad \rho_{F,\lambda}^*(F_p) := p^{2t} \rho_{F,\lambda}^{-1}(F_p).$$

Let α_i , $i = 1, 2, 3$ be the eigenvalues of $\rho_{F,\ell}(F_p)$, then the eigenvalues of $\rho_{F,\ell}^*(F_p)$ are $\beta_i := p^2/\alpha_i$. Since $\sum \alpha_i = a_p$, $\sum \alpha_i \alpha_j = pa_p$ (since now $\bar{a}_p = a_p$) and $\prod \alpha_i = p^3$, the sets of eigenvalues $\{\alpha_i\}$ and $\{\beta_i\}$ coincide.

Thus $L_p(\rho_F, s) = L_p(\rho_F^*, s)$ for all p not dividing N and so the (semi-simplifications of the) Galois representations are the same. It implies also that a subgroup of finite index of the image of $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ is contained in a group $G \subset \mathrm{GL}(W_\lambda)$ with $G \cong \mathrm{PGL}(2, K_\lambda)$. Examples of this are the Sym^2 of Galois representations in $\mathrm{GL}(2, \mathbb{Q}_\ell)$.

4.3

We will be especially interested in the non-selfdual case. Since we found several examples of Hecke eigenforms F with $K_F = \mathbb{Q}(i)$ we will consider that case here. To find corresponding Galois representations we use the fact that for any algebraic variety X defined over \mathbb{Q} , one has a Galois representation on the etale cohomology:

$$\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \longrightarrow \mathrm{GL}(H_{et}^n(X_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell)).$$

The point is to find a suitable X and (a subspace of) a suitable H_{et}^n . In case X is smooth, projective, and has good reduction mod p , theorems of Grothendieck and Deligne imply that the eigenvalue polynomial of F_p acting on $H_{et}^n(X_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell)$ has coefficients in \mathbb{Z} , is independent of ℓ and the eigenvalues of F_p have absolute value $p^{n/2}$.

The desired equality $L_p(F, s) = L_p(\rho_F, s)$ for the eigenforms F from (3.6) (and one expects the same more generally for certain cusp forms, ‘Ramanujan conjecture’), implies that the absolute value of the eigenvalues of $\rho_F(F_p)$ must be p . Therefore we will consider H_{et}^2 and take $\dim X > 1$ since $\dim H_{et}^2 = 1$ for curves.

A well-known theorem implies that $H_{et}^2(X_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell) \hookrightarrow H_{et}^2(S_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell)$ where S is a suitable surface contained in X . Thus we restrict ourselves to considering $H_{et}^2(S_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell)$ for a surface S .

The Galois representation on this \mathbb{Q}_ℓ -vector space is reducible in general, a decomposition is:

$$H_{et}^2(S_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell) = T_\ell \oplus \text{NS}(S_{\overline{\mathbb{Q}}}) \otimes_{\mathbb{Z}} \mathbb{Q}_\ell$$

where $\text{NS}(S_{\overline{\mathbb{Q}}})$ is the Néron-Severi group of the surface S over $\overline{\mathbb{Q}}$ (the Galois group permutes the classes of divisors modulo a Tate twist) and T_ℓ is the orthogonal complement of $\text{NS}(S_{\overline{\mathbb{Q}}})$ with respect to the intersection form. The intersection form is the cup product $H_{et}^2 \times H_{et}^2 \rightarrow H_{et}^4 \cong \mathbb{Q}_\ell$. The eigenvalues of Frobenius on $\text{NS}(S_{\overline{\mathbb{Q}}}) \otimes \mathbb{Q}$ are roots of unity multiplied by p , so $\rho_{F,\lambda}$, if it exists, should be a representation on a subspace of $T_\ell \otimes_{\mathbb{Q}_\ell} K_\lambda$.

In case T_ℓ has dimension 3, the Galois representation on it will be selfdual (due to the intersection form). To find a 3 dimensional Galois representations with $\text{trace} F_p \in \mathbb{Z}[i]$ as desired we assume that the surface has an automorphism, defined over \mathbb{Q} :

$$\phi : S \longrightarrow S, \quad \text{with} \quad \phi^4 = id_S.$$

Thus $\phi^* : H_{et}^2 \rightarrow H_{et}^2$ will commute with the Galois representation.

Assume moreover that $\dim T_\ell = 6$ and $\phi^* : T_\ell \rightarrow T_\ell$ has two 3-dimensional eigenspaces W_λ, W'_λ (with eigenvalue $\pm i$):

$$T_\lambda := T_\ell \otimes_{\mathbb{Q}_\ell} K_\lambda = W_\lambda \oplus W'_\lambda$$

with K_λ an extension of \mathbb{Q}_ℓ containing i . Then we have a 3-dimensional Galois representation σ' on W_λ . The determinant of $\sigma'(F_p)$ is in general not equal to p^3 but is $\chi(p)p^3$ for a Dirichlet character χ . Twisting σ' by this character we get a Galois representation

$$\sigma_{S,\lambda} : \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \longrightarrow \text{GL}(W_\lambda).$$

whose L -factors $L_p(\sigma_S, s)$ are similar to the $L_p(F, s)$ for the eigenforms in the example above.

Note that the intersection form (\cdot, \cdot) restricted to W_λ is trivial (it is invariant under pull-back by ϕ^* and extends K_λ -linearly: $(w_1, w_2) = (\phi^*w_1, \phi^*w_2) = (iw_1, iw_2) = i^2(w_1, w_2) = -(w_1, w_2)$ with $w_1, w_2 \in W_\lambda$). Thus there is no obvious reason for σ_S to be selfdual.

4.4

Now one has to search for such surfaces. The main problem is that in general $\dim H_{et}^2$ will be large but rank NS will be small. Thus it is not so easy to get $\dim T_\ell = 6$, see however [vG-T] and [vG-T2] for various examples.

The most interesting example is given by the one parameter family of surfaces S_a which are the smooth, minimal, projective model of the singular, affine surface defined in x, y, t -space by

$$t^2 = xy(x^2 - 1)(y^2 - 1)(x^2 - y^2 + axy), \quad \text{and} \quad (x, y, t) \longmapsto (y, -x, t)$$

defines the automorphism ϕ . In [vG-T], 3.7-3.9, we explain how to determine eigenvalue polynomials of $\sigma_{S,\lambda}(F_p)$, and thus the L -factors, basically using the Lefschetz trace formula and counting points on S over finite fields. The main result is:

4.5 Theorem.

([vG-T], 3.11; [GKTV], 3.9) The local L -factors of the modular forms for $N = 128, 160, 205$ in §3.6 are the same as the local L -factors of the Galois representations $\sigma_{S_a, \lambda}$, with $a = 2, 1, \frac{1}{16}$ respectively, for all odd primes $p \leq 173$ not dividing N .

4.6

In [vG-T2] we gave another construction of surfaces S which define 3 dimensional Galois representations. These surfaces are degree 4 cyclic base changes of elliptic surfaces $\mathcal{E} \rightarrow \mathbb{P}^1$. By taking the orthogonal complement to a large algebraic part in H_{et}^2 together with all cohomology coming from the intermediate degree 2 base change, one obtains a representation space, similar to T_ℓ , for $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$. Taking an eigenspace W_λ of the action of the automorphism of order 4 defining the cyclic base change finally gives 3 dimensional Galois representations.

Our main (technical) result is a formula for the traces of Frobenius elements on this space in terms of the number of points on \mathcal{E} and S over a finite field ([vG-T2], Theorem 3.4). This formula allows us to compute the characteristic polynomial of Frobenius in many cases.

We use this result to prove that certain examples yield selfdual representations, while others do not. For some of the selfdual cases we can actually exhibit 2-dimensional Galois representations (defined by elliptic curves) whose symmetric square seems to coincide with the 3-dimensional Galois representation.

We did not find new examples of non-selfdual Galois representations with the same local L -factors as modular forms, probably because the conductor of these Galois representations is too large. We would like to point out that there does not seem to be an explicit way to determine the conductor of the Galois representation σ_S in terms of the geometry of S (a surface over $\mathrm{Spec}(\mathbb{Q})$).

5 Non-cusp Forms and Galois Representations

5.1

In this section we give an example of the decomposition in Hecke eigenspaces of a cohomology group $H^3(\Gamma_0(N), \mathbb{C})$. We will take $N = 245$. This example is also mentioned in [GKTV], §3.5 where it is shown that a certain 8 dimensional Hecke invariant subspace of $H^3(\Gamma_0(245), \mathbb{C})$ contains no cusp forms. Here we extend this by interpreting most of the 83 dimensional space $H^3(\Gamma_0(245), \mathbb{C})$ in terms of so-called Eisenstein liftings of classical elliptic cusp forms and of Eisenstein series.

As before, if $F \in H^3(\Gamma_0(245), \mathbb{C})$ is an eigenform for all Hecke operators, we denote by K_F the field generated by all eigenvalues of the Hecke operators on F . As a first step towards the decomposition we have the following Proposition.

Proposition 1. *The cohomology group $H^3(\Gamma_0(245), \mathbb{C})$ decomposes as*

$$H^3(\Gamma_0(245), \mathbb{C}) = V_1 \oplus V_2 \oplus V_3 \oplus V_4 \oplus V_5$$

(as a module over the Hecke algebra), with

- $\dim V_1 = 25$ and V_1 is generated by eigenforms F with $K_F = \mathbb{Q}$;
- $\dim V_2 = 16$ and V_2 is generated by eigenforms F with $K_F = \mathbb{Q}(\sqrt{2})$;
- $\dim V_3 = 16$ and V_3 is generated by eigenforms F with $K_F = \mathbb{Q}(\sqrt{17})$;
- $\dim V_4 = 8$ and V_4 is generated by eigenforms F with $K_F = \mathbb{Q}(\sqrt{2}, \sqrt{-3})$;
- $\dim V_5 = 18$ and V_5 is generated by eigenforms F with $K_F = \mathbb{Q}(\sqrt{-3})$.

None of the spaces V_1, \dots, V_5 contains a non-zero cuspform; in fact, these spaces are generated by Eisenstein liftings or (in the case of V_4 and V_5) twists of such by cubic Dirichlet characters.

5.2

With notations as given in [GKTV] §3.5, one has $V_4 = V_a \oplus V_b$, hence this case of the above proposition is already described in *loc. sit.*

We briefly recall the two types of Eisenstein liftings of classical modular forms here. Let f be a normalized elliptic cuspform of level N and weight 2, which is an eigenform for the Hecke operators T_n with $(n, N) = 1$. Also, we allow f to be the normalized Eisenstein series of weight 2: $f = -B_2/4 + \sum_{n=1}^{\infty} \sigma_1(n)q^n$; so $a_p = p+1$, compare e.g. [Ko] for notations. The Fourier coefficients in the q -expansion $f = q + a_2q^2 + a_3q^3 + \dots$ define a Dirichlet series $L(f, s) = \sum_n a_n n^{-s}$. This series has an Euler product expansion with Euler factors $(1 - a_p p^{-s} + p^{1-2s})^{-1}$ for primes p which do not divide N (in case f is the Eisenstein series, these factors are $(1 - p^{-s})^{-1}(1 - p^{1-s})^{-1}$).

Given f , one constructs two eigenclasses $F_1, F_2 \in H^3(\Gamma_0(N), \mathbb{C})$. The F_1 has eigenvalue $pa_p + 1$ for the p th Hecke operator E_p , and F_2 eigenvalue $a_p + p^2$.

On the Galois side of the Langlands correspondence, it is relatively easy to describe these liftings. Namely, if f corresponds to a 2 dimensional λ -adic representation space V for $\text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$, then F_1 corresponds to $V(-1) \oplus \mathbb{Q}_{\lambda}(0)$ and F_2 to $V \oplus \mathbb{Q}_{\lambda}(-2)$ where $\mathbb{Q}_{\lambda}(n)$ is the 1 dimensional λ -adic representation space on which the Galois group acts by the $-n$ -th power of the cyclotomic character (thus F_p acts as p^{-n}). In case f is the Eisenstein series, we have $V = \mathbb{Q}_{\lambda}(0) \oplus \mathbb{Q}_{\lambda}(-1)$ and the two lifted representations coincide (both are $\mathbb{Q}_{\lambda}(0) \oplus \mathbb{Q}_{\lambda}(-1) \oplus \mathbb{Q}_{\lambda}(-2)$).

5.3

There exists a unique normalized cuspform of weight 2 and level 35 which has \mathbb{Q} -rational Fourier coefficients. This form yields 2 eigenclasses in $H^3(\Gamma_0(35), \mathbb{C})$; from the theory of oldforms [Ree], each of these appears three times at level $35 \cdot 7 = 245$.

Similarly, the modular form corresponding to the CM elliptic curve of conductor 49 gives rise to six oldforms which are Eisenstein liftings.

Starting from the Eisenstein series, one finds 7 forms at level 245 all with eigenvalues $1 + p + p^2$.

Finally, from tables of Cremona (as well as from unpublished tables of Cohen, Skoruppa and Zagier) it follows that there exist 3 (elliptic) newforms of level 245 which are Hecke eigenforms with rational eigenvalues. Each of them gives us two Eisenstein liftings.

Adding up, we now have $6 + 6 + 7 + 6 = 25$ eigenclasses of level 245 with rational eigenvalues. Our calculations made for the tables in [GKTV] revealed that, e.g., the Hecke operator E_2 has precisely 25 rational eigenvalues (counted with multiplicity). Hence the conclusion is, that the space V_1 given in Proposition 1 indeed has $\dim V_1 = 25$, and it is generated by Eisenstein liftings as claimed.

5.4

The cases V_2, V_3 are completely analogous. For V_2 , we note that there exist newforms of weight 2 and level 245 with q -expansion $q + \sqrt{2}q^2 + (1 + \sqrt{2})q^3 + \dots$ and $q + (1 + \sqrt{2})q^2 + (1 - \sqrt{2})q^3 + \dots$ respectively. These together with their Galois conjugate forms and their twists by the quadratic Dirichlet character modulo 7 give us 8 newforms of level 245. Each of them yields two Eisenstein liftings, and this precisely describes the space V_2 of dimension 16.

Similarly, there are exactly two (conjugate) newforms of level 35 with Fourier coefficients generating $\mathbb{Q}(\sqrt{17})$. They provide $2 \cdot 2 = 4$ Eisenstein liftings of level 35, and hence $3 \cdot 4 = 12$ oldforms of level 245. Twisting the newforms by the quadratic character modulo 7 yields newforms of level 245, and from these we find another 4 Eisenstein liftings. In this way, V_3 is generated.

5.5

Having described V_1, \dots, V_4 (the latter space was already treated in [GKTV]), and observing from Table 3.3 that $\dim H^3(\Gamma_0(245), \mathbb{C}) = 83$, we conclude we still have to describe a Hecke-invariant space of dimension $83 - (25 + 16 + 16 + 8) = 18$. To this end, we mention that at level $49 = 245/5$, our programs found a 6 dimensional Hecke invariant subspace on which the operator E_2 acts with 6 (pairwise conjugate, pairwise different) eigenvalues in $\mathbb{Q}(\sqrt{-3})$. Hence this space yields eigenforms with $K_F = \mathbb{Q}(\sqrt{-3})$. Moreover, it lifts to a Hecke invariant subspace of dimension $3 \cdot 6 = 18$ at level 245, which therefore exactly equals the summand V_5 of H^3 we did not describe yet.

As an example, the eigenvalues of the operator E_3 on V_5 are $a_3, \overline{a_3}, a_3\omega, \overline{a_3}\overline{\omega}, a_3\overline{\omega}$ and $\overline{a_3}\omega$ where $\omega^2 + \omega + 1 = 0$ and $a_3 = -5 - 3\sqrt{-3}$. This situation is explained as follows. The Euler factor that corresponds to a Hecke eigenclass is obtained using the polynomial $X^3 - a_p X^2 + pb_p X - p^3$, where a_p is the eigenvalue of the operator E_p . The number b_p similarly corresponds to the operator $D_p =$

T_{β_p} , defined using $\beta_p := \begin{pmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & 1 \end{pmatrix} \in \mathrm{GL}(3, \mathbb{Q})$. If the eigenclass is cuspidal,

then b_p is the complex conjugate of a_p . This in fact follows from the fact that the associated automorphic representation is unitary in that case. In our situation however, a computation shows that $b_3 = a_3 \neq \overline{a_3}$. Hence the representation cannot be unitary and therefore the eigenclasses here are not cuspidal.

Based on calculations for primes ≤ 131 , the Hecke eigenvalues seem to be as follows. For $p \neq 5, 7$ we have $b_p = a_p = \chi(p)(\psi(p) + p + \psi^2(p)p^2)$ with χ, ψ Dirichlet characters modulo 7 of order dividing 3. This corresponds to the sum of 1-dimensional Galois representations

$$(\chi\psi \otimes \mathbb{Q}_\lambda(0)) \oplus (\chi \otimes \mathbb{Q}_\lambda(-1)) \oplus (\chi\psi^2 \otimes \mathbb{Q}_\lambda(-2)).$$

6 Variations of Hodge Structures of Weight Two

6.1

In all our constructions for Galois representations we consider a subspace $T_\ell \subset H_{et}^2(S_{\overline{\mathbb{Q}}}, \mathbb{Q}_\ell)$. This subspace is defined using algebraic cycles, thus there exists also a Betti realization $T_{\mathbb{Z}} \subset H^2(S(\mathbb{C}), \mathbb{Z})$ (of the motive T) which is a polarized Hodge structure of weight two. We recall the relevant definitions and the main results of Griffiths and Carlson on the moduli of the $T_{\mathbb{Z}}$'s.

The main point is the essential difference with the weight one case (which is essentially the theory of abelian varieties). In the weight one case, one has a universal family of abelian varieties over suitable quotients of the Siegel space. In the weight two (and higher) case, the analogy of the Siegel space is a certain (subset of a) period domain, but in general (and in particular this is the case with the $T_{\mathbb{Z}}$ under consideration), the (polarized) Hodge structures obtained from algebraic varieties do not fill up the period space. In fact we will see that the Hodge structures like $T_{\mathbb{Z}}$ are parametrized by a 4-dimensional space, but those that come from geometry have at most a 2-dimensional deformation space (and imposing an automorphism of order 4 as we do implies a 1-dimensional deformation space).

It is not clear whether these period spaces (or the subvarieties parametrizing ‘geometrical’ Hodge structures) have good arithmetical properties like Shimura varieties.

6.2

Recall that a \mathbb{Z} -Hodge structure V of weight n is a free \mathbb{Z} -module of finite rank together with decomposition:

$$V_{\mathbb{C}} := V \otimes_{\mathbb{Z}} \mathbb{C} = \bigoplus_{p+q=n} V^{p,q}, \quad \text{with} \quad \overline{V^{p,q}} = V^{q,p},$$

where the $V^{p,q}$ are complex vector spaces and the bar indicates complex conjugation (given by $\overline{v \otimes z} = v \otimes \overline{z}$).

A rational Hodge structure $V_{\mathbb{Q}}$ is a finite dimensional \mathbb{Q} -vector space with a similar decomposition of $V_{\mathbb{C}} := V_{\mathbb{Q}} \otimes_{\mathbb{Q}} \mathbb{C}$. Thus a \mathbb{Z} -Hodge structure V determines a rational Hodge structure on $V_{\mathbb{Q}} := V \otimes_{\mathbb{Z}} \mathbb{Q}$.

A (rational) Hodge structure $V_{\mathbb{Q}}$ determines an \mathbb{R} -linear map, the Weil operator:

$$J : V_{\mathbb{R}} := V_{\mathbb{Q}} \otimes_{\mathbb{Q}} \mathbb{R} \longrightarrow V_{\mathbb{R}} \quad \text{with} \quad J_{\mathbb{C}} v_{p,q} = i^{p-q} v_{p,q}$$

for all $v_{p,q} \in V^{p,q}$ and $J_{\mathbb{C}}$ is the \mathbb{C} -linear extension of J . One has $J^2 = (-1)^n$ since $i^{2p-2q} = (-1)^{p-q} = (-1)^{p+q}$. Thus J determines a complex structure on $V_{\mathbb{R}}$ in case V has odd weight.

A polarization on a rational Hodge structure $V_{\mathbb{Q}}$ of weight n is a bilinear map

$$\Psi : V_{\mathbb{Q}} \times V_{\mathbb{Q}} \longrightarrow \mathbb{Q}, \quad \Psi_{\mathbb{C}}(v_{p,q}, v_{r,s}) = 0 \quad \text{unless} \quad p+r = q+s = n$$

(intrinsically: $\Psi : V_{\mathbb{Q}} \otimes V_{\mathbb{Q}} \rightarrow \mathbb{Q}(-n)$ is a morphism of Hodge structures) which satisfies the Riemann relations, that is, for all $v, w \in V_{\mathbb{R}}$:

$$\Psi(v, Jw) = \Psi(w, Jv), \quad \Psi(v, Jv) > 0 \quad (\text{if } v \neq 0)$$

thus Ψ defines an inner product $\Psi(-, J-)$ on $V_{\mathbb{R}}$.

One easily verifies, using the first property, that $\Psi(Jv, Jw) = \Psi(v, w)$, since also $\Psi(Jv, Jw) = \Psi(w, J^2v) = (-1)^n \Psi(w, v)$, a polarization is symmetric if n is even and antisymmetric if n is odd.

6.3

For a smooth complex projective variety X the cohomology groups $H^n(X, \mathbb{Q})$ are polarized rational Hodge structures of weight n . One writes $H^{p,q}(X) := H^n(X, \mathbb{C})^{p,q}$. In case X is a surface, the cup product on $H^2(X, \mathbb{Q})$ (note that $H^4(X, \mathbb{Q}) = \mathbb{Q}$) gives (-1 times) a polarization on the primitive cohomology H_{prim}^2 . In particular it induces a polarization Ψ on the sub-Hodge structure $T_{\mathbb{Q}} = \mathrm{NS}^{\perp}$ of $H^2(S(\mathbb{C}), \mathbb{Q})$ which we consider.

6.4

Let $T_{\mathbb{Z}}$ be a Hodge structure of weight 2 and rank 6 with

$$T_{\mathbb{C}} = T^{2,0} \oplus T^{1,1} \oplus T^{0,2}, \quad \dim T^{p,q} = 2$$

for all p, q . Then one easily verifies that:

$$T_{\mathbb{R}} = W_1 \oplus W_2 \quad \text{with} \quad \begin{cases} W_1 := T_{\mathbb{R}} \cap T^{1,1} \\ W_2 := T_{\mathbb{R}} \cap (T^{2,0} \oplus T^{0,2}) \end{cases}$$

For $v \in W_1 \subset T^{1,1}$ we have $Jv = v$ and thus $\Psi(v, v) = \Psi(v, Jv) > 0$, so Ψ is positive definite on W_1 . Hence we can choose an \mathbb{R} basis f_1, f_2 of W_1 which is orthonormal w.r.t. Ψ and which is a \mathbb{C} -basis of $T^{1,1} = W_1 \otimes_{\mathbb{R}} \mathbb{C}$.

For $v \in W_2$ we have $v = v_{2,0} + v_{0,2}$ thus $Jv = -v$ and so Ψ is negative definite on W_2 . Let $v_1 := e_1 + \bar{e}_1$, $v_2 := e_2 + \bar{e}_2$ be an orthonormal basis for $(-1/2)\Psi$ on W_2 with $e_1, e_2 \in V^{2,0}$. Then e_1, e_2 is a \mathbb{C} -basis of $T^{2,0}$ (and thus \bar{e}_1, \bar{e}_2 is a \mathbb{C} -basis of $T^{0,2}$). Note $-2 = \Psi(e_1 + \bar{e}_1, e_1 + \bar{e}_1) = \Psi(e_1, \bar{e}_1) + \Psi(\bar{e}_1, e_1) = 2\Psi(e_1, \bar{e}_1)$ (since Ψ is symmetric). In this way one finds $\Psi(e_k, \bar{e}_l) = -\delta_{kl}$ (Kronecker's delta) thus $\Psi_{\mathbb{C}}$ is given by the matrix Q on the basis $e_1, e_2, f_1, f_2, \bar{e}_1, \bar{e}_2$ of $T_{\mathbb{C}}$:

$$Q = \begin{pmatrix} 0 & 0 & -I \\ 0 & I & 0 \\ -I & 0 & 0 \end{pmatrix}.$$

6.5

We consider first order deformations of the polarized Hodge structure $T_{\mathbb{Z}}$ as in §6.4. Thus we fix the \mathbb{Z} -module and the bilinear map Ψ and consider deformations of the Hodge structure induced by deformations of an algebraic variety X with $T_{\mathbb{Z}} \subset H^2(X, \mathbb{Z})$, that is, of the direct sum decomposition $T_{\mathbb{C}} = \bigoplus T^{p,q}$.

The first order deformations of a smooth complex projective algebraic variety X are parametrized by $H^1(X, \Theta_X)$ with Θ_X the tangent bundle of X (Kodaira-Spencer theory). The isomorphisms $H^{p,q}(X) = H^q(X, \Omega^p)$ and the contraction map $\Theta_X \otimes_{\mathcal{O}_X} \Omega_X^p \rightarrow \Omega_X^{p-1}$ give a cup product map:

$$H^1(X, \Theta_X) \otimes H^{p,q}(X) \longrightarrow H^{p-1, q+1}(X).$$

Thus, for any n , we obtain a map, called the infinitesimal period map:

$$\delta : H^1(X, \Theta_X) \longrightarrow \bigoplus_{p+q=n} \text{Hom}(H^{p,q}(X), H^{p-1, q+1}(X)).$$

Griffiths proved that for $\theta \in H^1(X, \Theta_X)$, the deformation of the Hodge structure induced by the deformation of X in the direction of θ is essentially given by $\delta(\theta)$.

The subspace $\mathfrak{S}(\delta)$ of $\bigoplus_{p+q=n} \text{Hom}(H^{p,q}(X), H^{p-1, q+1}(X))$ satisfies (at least) two conditions. The first comes from the polarization (see §6.6), the second is an integrability condition found by Griffiths which is non-trivial only if the weight of the Hodge structure is greater than one (see §6.8).

We will now spell out the restriction of these conditions to the sub Hodge structure $\bigoplus_{p+q=n} \text{Hom}(T^{p,q}, T^{p-1, q+1})$.

6.6

The condition that $\psi \in \bigoplus_{p+q=n} \text{Hom}(T^{p,q}, T^{p-1, q+1}) \subset \text{End}(T_{\mathbb{C}})$ preserves the polarization on T , is that $\Psi((I + t\psi)v, (I + t\psi)w) = \Psi(v, w)$ when $t^2 = 0$:

$$\Psi_{\mathbb{C}}(\psi(v), w) + \Psi_{\mathbb{C}}(v, \psi(w)) = 0 \quad \forall x, y \in T_{\mathbb{C}}$$

This condition implies that if ψ preserves Ψ , then it is determined by ψ_2 where

$$\psi = (\psi_2, \psi_1) \in \text{Hom}(T^{2,0}, T^{1,1}) \oplus \text{Hom}(T^{1,1}, T^{0,2}).$$

In fact, for all $v \in T^{2,0}$ and $w \in T^{1,1}$ we have: $\Psi_{\mathbb{C}}(v, \psi_1(w)) = -\Psi_{\mathbb{C}}(\psi_2(v), w)$. Since $\Psi_{\mathbb{C}}$ identifies $(T^{0,2})^{\text{dual}}$ with $T^{2,0}$, this equality thus defines $\phi_1(w)$ in terms of ϕ_2 .

6.7

With respect to the basis of $T_{\mathbb{C}}$ considered in 6.4, $\psi \in \mathrm{Hom}(T^{2,0}, T^{1,1}) \oplus \mathrm{Hom}(T^{1,1}, T^{0,2}) \subset \mathrm{End}(T_{\mathbb{C}})$ is given by a matrix N and the condition on ψ becomes ${}^t N Q + Q N = 0$ so:

$$N = \begin{pmatrix} 0 & 0 & 0 \\ A & 0 & 0 \\ 0 & B & 0 \end{pmatrix} \quad \text{and} \quad B = {}^t A$$

where the matrix A (defining $\phi_2 : T^{2,0} \rightarrow T^{1,1}$) can be chosen arbitrarily. This gives an isomorphism between the space $M_2(\mathbb{C})$ of 2×2 complex matrices and polarization preserving deformations ψ :

$$M_2(\mathbb{C}) \xrightarrow{\cong} (\mathrm{Hom}(T^{2,0}, T^{1,1}) \oplus \mathrm{Hom}(T^{1,1}, T^{0,2}))_{\psi}, \quad A \longmapsto N(A) := \begin{pmatrix} 0 & 0 & 0 \\ A & 0 & 0 \\ 0 & {}^t A & 0 \end{pmatrix}.$$

Thus we have a four dimensional deformation space. In case of Hodge structures of weight one, preserving the polarization is the only infinitesimal condition. Here, in the weight two case, there is however another condition.

6.8

An important restriction, discovered by Griffiths, on the image of δ is:

$$[\mathrm{Im}\delta, \mathrm{Im}\delta] = 0 \quad \text{i.e.} \quad \delta(\alpha) \circ \delta(\beta) = \delta(\beta) \circ \delta(\alpha),$$

for all $\alpha, \beta \in H^1(X, \Theta_X)$, so $\Im(\delta)$ is an abelian subspace of $\mathrm{End}(T_{\mathbb{C}})$. For Hodge structures of weight $n \geq 2$ this imposes non-trivial conditions on the (dimension of) the image of δ . We consider again our example (cf. [Ca]).

6.9

We already determined the polarization preserving deformations in §6.7. Using the same notation we find that Griffiths' condition is:

$$N(A)N(B) = N(B)N(A) \quad \text{thus} \quad {}^t AB = {}^t BA.$$

This condition can be rephrased as saying that ${}^t AB$ must be symmetric.

Thus the image of δ is at most two dimensional and if it is two dimensional with basis $N(A), N(B)$ then A and B span a maximal isotropic subspace of the symplectic form:

$$E : M_2(\mathbb{C}) \times M_2(\mathbb{C}) \longrightarrow \mathbb{C}, \quad E(A, B) := a_{11}b_{12} - a_{12}b_{11} + a_{21}b_{22} - a_{22}b_{21} = 0.$$

We recall that we also have an automorphism $\phi^* : T \rightarrow T$, preserving this automorphism gives another non-trivial condition on the deformations. Thus the one parameter in our surfaces S_a (and in the other examples from [vG-T2]) is the maximal possible.

References

- AGG. A. Ash, D. Grayson and P. Green, Computations of Cuspidal Cohomology of Congruence Subgroups of $\mathrm{SL}(3, \mathbb{Z})$, *J. Number Theory* **19** (1984) 412–436.
- AR. A. Ash and L. Rudolph, The modular symbol and continued fractions in higher dimensions, *Invent. Math.* **55** (1979) 241–250.
- Ca. J. A. Carlson, Bounds on the dimension of variations of Hodge structure, *Trans. A.M.S.* **294** (1986) 45–64, Erratum: *Trans. A.M.S.* **299** (1987) 429.
- C1. L. Clozel, Motifs et formes automorphes: applications du principe de fonctorialité. In: L. Clozel and J.S. Milne (eds.), *Automorphic Forms, Shimura Varieties and L-functions*, Proceedings of the Ann Arbor Conference, 77–159. New York London: Academic Press (1990).
- C2. L. Clozel, Représentations galoisiennes associées aux représentations automorphes autoduales de $\mathrm{GL}(n)$. *Publ. Math. IHES* **73** (1991) 79–145.
- D. P. Deligne, Formes modulaires et représentations ℓ -adiques. In: *Sém. Bourbaki*, Springer LNM **179** (1971) 136–186.
- vG-T. B. van Geemen and J. Top, A non-selfdual automorphic representation of GL_3 and a Galois representation, *Invent. Math.* **117** (1994) 391–401.
- vG-T2. B. van Geemen and J. Top, Selfdual and non-selfdual 3-dimensional Galois representations, *Compos. Math.* **97** (1995) 51–70.
- GKTV. B. van Geemen, W. van der Kallen, J. Top and A. Verberkmoes, Hecke eigenforms in the Cohomology of Congruence Subgroups of $\mathrm{SL}(3, \mathbb{Z})$, *Experimental Math.* **6** (1997) 163–174.
- Ko. N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*. Springer-Verlag, New York etc.: 1984.
- Ree. M. Reeder, Old forms on $\mathrm{GL}(n)$, *Am. J. of Math.* **113** (1991) 911–930.

Modular Symbols and Hecke Operators

Paul E. Gunnells

Columbia University, New York, NY 10027, USA

Abstract. We survey techniques to compute the action of the Hecke operators on the cohomology of arithmetic groups. These techniques can be seen as generalizations in different directions of the classical modular symbol algorithm, due to Manin and Ash-Rudolph. Most of the work is contained in papers of the author and the author with Mark McConnell. Some results are unpublished work of Mark McConnell and Robert MacPherson.

1 Introduction

1.1

Let G be a semisimple algebraic group defined over \mathbb{Q} , and let $\Gamma \subset G(\mathbb{Q})$ be an arithmetic subgroup. The cohomology of Γ plays an important role in number theory, through its connection with automorphic forms and representations of the absolute Galois group $\text{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$. This relationship is revealed in part through the action of the *Hecke operators* on the complex cohomology $H^*(\Gamma; \mathbb{C})$. These are endomorphisms induced from a family of correspondences associated to the pair $(\Gamma, G(\mathbb{Q}))$; the arithmetic nature of the cohomology is contained in the eigenvalues of these linear maps.

For $\Gamma \subset \text{SL}_n(\mathbb{Z})$, the *modular symbols* and *modular symbol algorithm* of Manin [17] and Ash-Rudolph [8] provide a concrete method to compute the Hecke eigenvalues in $H^\nu(\Gamma; \mathbb{C})$, where $\nu = n(n+1)/2 - 1$ is the top degree (§2). These symbols have allowed many researchers to fruitfully explore the number-theoretic significance of this cohomology group, especially for $n = 2$ and 3 [3,7,5,21,22]. For all their power, though, modular symbols have limitations:

- The group G must be the linear group SL_n .
- The cohomology must be in the top degree ν .
- The group Γ must be a subgroup of $\text{SL}_n(\mathbb{Z})$, or more generally $\text{SL}_n(R)$, where R is a Euclidean ring of integers of a number field.

1.2

In this article we discuss new techniques to compute the Hecke action on the cohomology of arithmetic groups that can be seen as generalizing the modular symbol algorithm by relaxing the three restrictions above. First in §3 we relax the first restriction by replacing the linear group SL_n with the symplectic group Sp_{2n} [14]. Next in §4, we relax the second restriction and consider computations

in $H^{\nu-1}(\Gamma)$, where $\Gamma \subset \mathrm{SL}_n(\mathbb{Z})$ and $n \leq 4$ [13]. Finally, in the last two sections we relax all three restrictions, and consider arithmetic groups associated to *self-adjoint homogeneous cones* (§5) [12,15], and arithmetic groups for which a *well-rounded retract* is defined (§6) [16]. The first class includes $\mathrm{SL}_n(\mathcal{O}_K)$, where \mathcal{O}_K is the maximal order of a totally real or CM field, as well as arithmetic groups associated to the positive-definite 3×3 Hermitian octavic matrices. The second class includes arithmetic subgroups of $\mathrm{SL}_n(D)$, where D is a division algebra over \mathbb{Q} .

Most of this work is contained in papers of the author [14,12,13] or the author in joint work with Mark McConnell [15]. The last section is a summary of unpublished results of Robert MacPherson and Mark McConnell [16]. We have omitted other work, notably that of Bygott [10], Teitelbaum [20], and Merel [18], because of lack of space and/or author's expertise. It is a pleasure to thank Avner Ash, Robert MacPherson, and Mark McConnell for many conversations about these topics.

2 Classical Modular Symbols

2.1

We begin by recalling the classical modular symbol algorithm following Ash-Rudolph [8]. For simplicity we consider subgroups of $\mathrm{SL}_n(\mathbb{Z})$, although everything we say can be generalized to subgroups of $\mathrm{SL}_n(R)$, where R is a Euclidean maximal order in a number field (cf. [11]).

Let $\Gamma \subset \mathrm{SL}_n(\mathbb{Z})$ be a torsion-free finite-index subgroup, and let $m \in M_n(\mathbb{Q})$, the $n \times n$ matrices over \mathbb{Q} . We want to show how to use m to construct a class in $H^\nu(\Gamma)$. To this end, let X be the symmetric space $\mathrm{SL}_n(\mathbb{R})/\mathrm{SO}(n)$, let \bar{X} be the bordification constructed by Borel-Serre [9], and let $\partial\bar{X} = \bar{X} \setminus X$. Let $M = \Gamma \backslash X$, $\bar{M} = \Gamma \backslash \bar{X}$, and $\partial\bar{M} = \bar{M} \setminus M$. Then \bar{M} is a smooth manifold with corners, and $H^*(\Gamma) \cong H^*(\bar{M})$. We have an exact sequence

$$H_{n-1}(\partial\bar{X}) \rightarrow H_n(\bar{X}, \partial\bar{X}) \rightarrow H_n(\bar{M}, \partial\bar{M}) \rightarrow H^\nu(\bar{M}) \quad (1)$$

coming from the sequence of the pair $(\partial\bar{X}, \bar{X})$, the canonical projection $\bar{X} \rightarrow \bar{M}$, and Lefschetz duality. Moreover, the boundary $\partial\bar{X}$ has the homotopy type of the *Tits building* $\mathcal{B} = \mathcal{B}_{\mathrm{SL}}$ associated to $\mathrm{SL}_n(\mathbb{Q})$. This is an $(n-1)$ -dimensional simplicial complex whose k -simplices Δ are in bijection with flags F of rational subspaces

$$F = \{0 \subsetneq F_1 \subsetneq \cdots \subsetneq F_{k+1} \subsetneq \mathbb{Q}^n\};$$

we have $\Delta \subset \Delta'$ if and only if $F \subset F'$.

Any ordered tuple of nonzero rational vectors determines a maximal rational flag by defining F_k to be the span of the first k vectors. Hence if $m \in M_n(\mathbb{Q})$ has nonzero columns, the different orderings of the columns determine $n!$ different oriented $(n-1)$ -simplices in \mathcal{B} . These simplices can be thought of as an oriented simplicial cycle giving a class $[m] \in H_{n-1}(\mathcal{B}) \cong H_{n-1}(\partial\bar{X})$. The class $[m]$ is

called a *modular symbol*, and these classes span $H_{n-1}(\mathcal{B})$. According to Ash-Rudolph, the map $\Phi: H_{n-1}(\mathcal{B}) \rightarrow H^\nu(\Gamma)$ induced by (1) is surjective; hence the (duals of) the modular symbols span $H^\nu(\Gamma)$.

2.2

Write $[m] = [m_1, \dots, m_n]$, where each column $m_i \in \mathbb{Q}^n \setminus \{0\}$, and let \mathcal{M}_n be the \mathbb{Z} -module generated by the classes of the symbols $[m]$. Using the description in §2.1, one can show that elements of \mathcal{M}_n satisfy the following relations:

1. $[qm_1, m_2, \dots, m_n] = [m]$, for $q \in \mathbb{Q}^\times$.
2. $[m_{\sigma(1)}, \dots, m_{\sigma(n)}] = \text{sgn}(\sigma)[m]$, for any permutation σ .
3. $[m] = 0$ if $\det m = 0$.
4. $\sum_{i=0}^n (-1)^i [m_0, \dots, \hat{m}_i, \dots, m_n] = 0$, for any $n+1$ vectors m_0, \dots, m_n (the “cocycle relation”).

By the first relation, \mathcal{M}_n is generated by those $[m]$ such that m_i is integral and primitive for all i . If $m \in \text{SL}_n(\mathbb{Z})$, then $[m]$ is called a *unimodular symbol*. We have the following fundamental result of Manin ($n = 2$) and Ash-Rudolph ($n \geq 2$):

Theorem 1. [17,8] *Any modular symbol is homologous to a finite sum of unimodular symbols.*

We sketch the proof. If $|\det m| > 1$, then one can show there exists $v \in \mathbb{Z}^n \setminus \{0\}$ such that

$$0 \leq |\det m_i(v)| < |\det m|, \quad \text{for } i = 1, \dots, n. \quad (2)$$

where $m_i(v)$ is the matrix obtained by replacing the column m_i with v . Such a v is called a *reducing point* for m . Then applying the cocycle relation to the tuple v, m_1, \dots, m_n yields an expression for $[m]$ in terms of the symbols $[m_i(v)]$. By induction this completes the proof.

This process of rewriting a modular symbol as a sum of unimodular symbols is called the *modular symbol algorithm*. Using this algorithm one can compute the action of the Hecke operators on $H^\nu(\Gamma)$ as follows. There are only finitely many unimodular symbols mod Γ , and from them one can select a subset dual to a basis of $H^\nu(\Gamma)$. A Hecke operator acts on the modular symbols by taking a unimodular symbol into a sum of nonunimodular symbols. Hence the modular symbol algorithm allows one to compute the Hecke action on a basis, from which one can easily compute the eigenvalues.

3 Symplectic Modular Symbols

3.1

For the first generalization we replace the linear group with the symplectic group [14]. Let V be a $2n$ -dimensional \mathbb{Q} -vector space with basis $\{e_1, \dots, e_n, e_{\bar{n}}, \dots, e_{\bar{1}}\}$,

where $\bar{i} := 2n + 1 - i$. Let $\langle \cdot, \cdot \rangle: V \times V \rightarrow \mathbb{Q}$ be the nondegenerate, alternating bilinear form defined by

$$\langle e_i, e_j \rangle = \begin{cases} 1 & \text{if } j = \bar{i} \text{ with } i < j \\ -1 & \text{if } j = \bar{i} \text{ with } i > j \\ 0 & \text{otherwise.} \end{cases}$$

The form $\langle \cdot, \cdot \rangle$ is called a *symplectic form*, and the *symplectic group* $\mathrm{Sp}_{2n}(\mathbb{Q})$ is defined to be the subgroup of $\mathrm{SL}_{2n}(\mathbb{Q})$ preserving $\langle \cdot, \cdot \rangle$.

3.2

Much of §2 carries over without change, but there are some new wrinkles coming from the geometry of the symplectic form. Recall that an *isotropic subspace* is one on which the symplectic form vanishes, and that maximal (necessarily n -dimensional) isotropic subspaces are called *Lagrangian*. Then the symplectic building $\mathcal{B}_{\mathrm{Sp}}$ has a k -simplex for every length $(k+1)$ flag of isotropic subspaces. Since the columns of a symplectic matrix m satisfy

$$\langle m_i, m_j \rangle = 0 \quad \text{if and only if} \quad i \neq \bar{j}, \tag{3}$$

it is easy to see that m determines $2^n \cdot n!$ oriented simplices of maximal dimension in $\mathcal{B}_{\mathrm{Sp}}$.

Furthermore, the arrangement of these simplices in $\mathcal{B}_{\mathrm{Sp}}$ differs from the linear case. Suppose we use the columns of m to induce points in the projective space $\mathbb{P}^{2n-1}(\mathbb{Q})$. Then the Lagrangian subspaces spanned by the columns of m become $(n-1)$ -dimensional flats arranged in the configuration of a *hyperoctahedron*.¹ This time m determines a class $[m] \in H_{n-1}(\mathcal{B}_{\mathrm{Sp}})$, and as m ranges over all rational matrices with columns satisfying (3), the duals of the classes $[m]$ span $H^\nu(\Gamma)$.

3.3

As a first step towards a symplectic modular symbol algorithm, one must understand the analogues of the relations from §2.2. The analogues of 1–3 are only slightly different to reflect the hyperoctahedral symmetry. The cocycle relation, however, is more interesting. A symbol $[m]$ and a generic nonzero rational point $v \in V$ determine $2n$ modular symbols $[m_i(v)]$ as follows. For any pair (i, j) with $i \neq \bar{j}$, we define points m_{ij} by

$$m_{ij} := \langle v, m_j \rangle m_i - \langle v, m_i \rangle m_j.$$

Let $[m_i(v)]$ be the modular symbol obtained by replacing $m_{\bar{i}}$ with v , and replacing the m_j with $j \notin \{i, \bar{i}\}$ by m_{ij} . Then one can show $[m] = \sum \varepsilon_i [m_i(v)]$ for appropriate signs ε_i .

For an example of this relation, consider Figure 1. The figure on the left shows the cocycle relation for Sp_4 in terms of a configuration in \mathbb{P}^3 . The black dots are the points corresponding to the m_i , the grey dot correspond to v , and the triangles to the points m_{ij} .

¹ Recall that a hyperoctahedron is the convex hull of the $2n$ points $\{\pm e \mid e \in E\}$, where E is the standard basis of \mathbb{R}^{2n} .

3.4

Now we can describe the symplectic modular symbol algorithm. Let $m \in M_{2n}(\mathbb{Z})$ have columns satisfying (3). Then $\det m = \prod_{i=1}^n \langle m_i, m_{\bar{i}} \rangle$, and one can show that if $|\det m| > 1$, there exists a vector $v \in \mathbb{Z}^n \setminus \{0\}$ such that

$$0 \leq |\langle m_i, v \rangle| < \langle m_i, m_{\bar{i}} \rangle, \quad \text{for } i = 1, \dots, 2n.$$

We can apply v to $[m]$ in the cocycle relation alluded to in §3.3, but we will unfortunately find that $|\det m_i(v)| > |\det m|$ in general. However, all is not lost. It turns out that for fixed i and fixed v , the $2n - 2$ vectors $\{m_{ij} \mid j \neq i, \bar{i}\}$ form a tuple that can be regarded as a symplectic modular symbol associated to Sp_{2n-2} . By induction one knows how to make these symbols unimodular, and this allows one to further reduce the $[m_i(v)]$ (cf. the right of Figure 1).

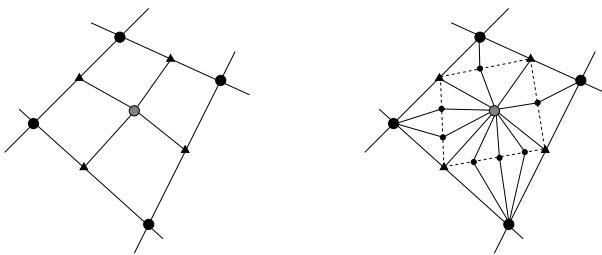


Fig. 1. $G = \mathrm{Sp}_4$. On the left, the outer square is the original symbol $[m]$, and the four smaller squares are the symbols $[m_i(v)]$. On the right, each modular symbol has been further reduced by applying the modular symbol algorithm to $\mathrm{Sp}_2 = \mathrm{SL}_2$ modular symbols.

4 Below the Cohomological Dimension

4.1

We return to the case of SL_n . As said before, a limitation of the modular symbol algorithm is that one can compute the Hecke action only on the top degree cohomology. For $n \leq 3$ this cohomology group is very interesting: it contains *cuspidal* classes, i.e. classes associated to cuspidal automorphic forms. If $n \geq 4$, however, the top degree cohomology group no longer contains cuspidal classes. In particular, if $n = 4$, one is really interested in computing the Hecke action on $H^5(\Gamma)$. For instance, recent work Jasper Scholten has constructed 4-dimensional representations of $\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$ that should be related to a cuspidal Hecke eigenform in $H^5(\Gamma)$ for some $\Gamma \subset \mathrm{SL}_4(\mathbb{Z})$ [19]. The modular symbol algorithm, however, applies only to $H^6(\Gamma)$.

In this section we describe an algorithm that for $n \leq 4$ allows computation of the Hecke action on $H^{\nu-1}(\Gamma)$ [13]. However, there is one caveat: we cannot

prove the algorithm will terminate. In practice, happily, the algorithm has always converged, and has permitted investigation of this cohomology [4].

4.2

To compute with lower degree cohomology groups, we use the *sharbly complex* S_* [2]. For $k \geq 0$, let S_k be the $\mathbb{Z}\Gamma$ -module generated by the symbols $\mathbf{u} = [v_1, \dots, v_{n+k}]$, where $v_i \in \mathbb{Q} \setminus \{0\}$, modulo the analogues of relations 1–3 in §2.2. Elements of S_k are called k -sharbles. Let $\partial: S_k \rightarrow S_{k-1}$ be the map $\mathbf{u} \mapsto \sum_i (-1)^i [v_1, \dots, \widehat{v}_i, \dots, v_{n+k}]$, linearly extended to all of S_k . There is a map $S_0 \rightarrow \mathcal{M}_n$ giving a $\mathbb{Z}\Gamma$ -free resolution of \mathcal{M}_n , and one can show that this implies $H^{\nu-k}(\Gamma; \mathbb{C}) \cong H_k(S_* \otimes \mathbb{C})$.

As in §2.2, it suffices to consider k -sharbles $\mathbf{u} = [v_1, \dots, v_{n+k}]$ with all v_i integral and primitive. Any modular symbol of the form $[v_{i_1}, \dots, v_{i_n}]$, where $\{i_1, \dots, i_n\} \subset \{1, \dots, n+k\}$, is called a *submodular symbol* of \mathbf{u} .

Let $\xi = \sum n(\mathbf{u})\mathbf{u}$ be a sharbly chain. We denote by $\|\xi\|$ the maximum absolute value of the determinant of any submodular symbol of ξ . The chain ξ is called *reduced* if $\|\xi\| = 1$. It is known that reduced 1-sharbly cycles provide a finite spanning set of $H^{\nu-1}(\Gamma; \mathbb{C})$ for $n \leq 4$.

Since the Hecke operators take reduced sharbly cycles to nonreduced cycles, our goal is to apply the modular symbol algorithm *simultaneously* over a nonreduced 1-sharbly cycle ξ to lower the determinants of the submodular symbols. Hence we are faced with two problems: first, how do we combine reducing points with the original 1-sharbly ξ to produce a new 1-sharbly ξ' homologous to ξ ; second, how do we choose the reducing points so that $\|\xi'\| < \|\xi\|$?

4.3

To address the first issue we do the following. Suppose $\mathbf{u} = [v_1, \dots, v_{n+1}]$ satisfies $n(\mathbf{u}) \neq 0$, and for $i = 1, \dots, n+1$, let \mathbf{v}_i be the submodular symbol $[v_1, \dots, \widehat{v}_i, \dots, v_{n+1}]$. Assume that all these submodular symbols are nonunimodular, and for each i let w_i be a reducing point for \mathbf{v}_i .

For any subset $I \subset \{1, \dots, n+1\}$, let \mathbf{u}_I be the 1-sharbly $[u_1, \dots, u_{n+1}]$, where $u_i = w_i$ if $i \in I$, and $u_i = v_i$ otherwise. Then we have a relation in S_1 given by

$$\mathbf{u} = - \sum_{I \neq \emptyset} (-1)^{\#I} \mathbf{u}_I. \quad (4)$$

Geometrically this relation can be expressed using the combinatorics of the hyperoctahedron [13, §4.4]. More generally, if some \mathbf{v}_i happen to be unimodular, then one can construct a similar relation using an iterated cone on a hyperoctahedron.

4.4

Now we apply the construction in §4.3 to all the 1-sharbles \mathbf{u} with $n(\mathbf{u}) \neq 0$, and we choose reducing points Γ -equivariantly. Specifically, if \mathbf{v} and \mathbf{v}' are

two submodular symbols of ξ with $\gamma\mathbf{v} = \mathbf{v}'$, then we choose the corresponding reducing points such that $\gamma w = w'$. After applying (4) to all the \mathbf{u} we determine a new 1-sharblly cycle ξ' . Clearly ξ' is homologous to ξ . We claim that $\|\xi'\|$ should be less than $\|\xi\|$.

To see why this should be true, consider the 1-sharblies \mathbf{u}_I on the right of (4). Of these 1-sharblies, those with $\#I = 1$ contain the \mathbf{v}_i among their submodular symbols. We claim that since ξ is a *cycle* mod Γ , and since the reducing points were chosen Γ -equivariantly over ξ , these 1-sharblies will not appear in ξ' . Hence by construction we have eliminated some of the “bad” submodular symbols from ξ .

4.5

Unfortunately, this doesn’t guarantee that $\|\xi'\| < \|\xi\|$. The problem is that we have no way of knowing that the submodular symbols of the \mathbf{u}_I with $\#I > 1$ don’t have large determinants. Indeed, this brings us back to the second question raised in §4.2, since if the reducing points are chosen naïvely, these submodular symbols *will* have large determinants. However, we claim that one can (conjecturally) choose the reducing points “uniformly” over ξ in a sense by using LLL-reduction, and that this problem doesn’t occur in practice. In fact, in thousands of computer tests and in applications, we have always found $\|\xi'\| < \|\xi\|$ if $n \leq 4$ and $\|\xi\| > 1$. We refer the interested reader to [13] for details.

5 Self-Adjoint Homogeneous Cones

5.1

Now we describe a different approach to computing the Hecke action that can be found in [12,15]. The main idea is to replace modular symbols and sharblly chains with chains built from rational polyhedral cones, and to replace “uni-modularization” with moving the support of a chain into a certain canonically defined set of rational polyhedral cones. The results of this section apply to any arithmetic group that is associated to a *self-adjoint homogeneous cone*; the reduction theory in this generality is due to Ash [6, Ch. 2]. However, for simplicity we describe the results in the context of Voronoï’s work reduction theory of real positive-definite quadratic forms [23].

Let V be the real vector space of all real symmetric $n \times n$ matrices, and let C be the subset of positive-definite matrices. Then C is a cone, i.e. C is a convex set closed under homotheties and containing no straight line. The group $\mathrm{SL}_n(\mathbb{Z})$ acts on V preserving C , and the action commutes with homotheties. In fact, modulo homotheties C is isomorphic to $X = \mathrm{SL}_n(\mathbb{R})/\mathrm{SO}(n)$; this exhibits a hidden linear structure of the symmetric space X .

Let \bar{C} be the closure of C in V . Voronoï showed how to construct a set \mathcal{V} of rational polyhedral cones in \bar{C} such that

1. Γ acts on \mathcal{V} .
2. If $\sigma \in \mathcal{V}$ then so is any face of σ .
3. If $\sigma, \tau \in \mathcal{V}$, then $\sigma \cap \tau$ is a face of each.
4. Modulo Γ , the set \mathcal{V} is finite.
5. The intersections $\sigma \cap C$ cover C .

The cones \mathcal{V} provide a reduction theory for C in the following sense: any $x \in C$ lies in a unique cone $\sigma(x) \in \mathcal{V}$, and the number of $\gamma \in \Gamma$ such that $\gamma \cdot \sigma(x) = \sigma(x)$ is bounded. Given $x \in C$, there is an explicit algorithm, the *Voronoi reduction algorithm*, to find $\sigma(x)$.

The Voronoi cones descend modulo homotheties to induce a decomposition of X into cells. Furthermore, we can enlarge C to a cone \tilde{C} such that, if \tilde{X} denotes \tilde{C} modulo homotheties, then the quotient $\Gamma \backslash \tilde{X}$ is compact. This *Satake compactification* of $\Gamma \backslash X$ is singular in general, but nevertheless can still be used to compute $H^*(\Gamma; \mathbb{C})$. For us, the salient points are that the images of the Voronoi cones induce a decomposition of \tilde{C} , with all the properties listed above, and that the Voronoi reduction algorithm extends to the boundary $\partial \tilde{C} := \tilde{C} \setminus C$.

5.2

Now let \mathbf{C}_*^R be the complex over \mathbb{C} generated by *all* simplicial rational polyhedral cones in \tilde{C} , and let \mathbf{C}_*^V be the subcomplex generated by Voronoi cones.² For any chain $\xi \in \mathbf{C}_*^R$, let $\text{supp } \xi$ be the set of cones supporting ξ . The complex \mathbf{C}_*^R is analogous to the sharbly complex, and the subcomplex \mathbf{C}_*^V to the subcomplex generated by the reduced sharblies. In general, however, \mathbf{C}_*^V is not isomorphic to the complex of reduced sharblies. Cycles $\xi \in \mathbf{C}_*^V$ can be used to compute $H^*(\Gamma)$, but the image $T(\xi)$ of ξ under a Hecke operator won't be supported on Voronoi cones. Hence we must show how to push $T(\xi)$ back into \mathbf{C}_*^V .

To accomplish this we have essentially two tools—we can subdivide the cones in $\text{supp } T(\xi)$, and we can use the Voronoi reduction algorithm to determine the cone any point lies in. We apply these as follows. Using the linear structure on \tilde{C} , we first subdivide $T(\xi)$ very finely into a chain ξ' . Then to each 1-cone $\tau \in \text{supp } \xi'$, we assign a 1-cone $\rho_\tau \in \partial \tilde{C}$, and we use the combinatorics of ξ' to assemble the ρ_τ into a cycle ξ'' homologous to ξ . We claim that if ξ' is constructed so that 1-cones $\tau \in \text{supp } \xi'$ lie in the *same or adjacent Voronoi cones*, then the ρ_τ can be chosen to ensure $\xi'' \in \mathbf{C}_*^V$.

5.3

We illustrate this process for SL_2 ; more details can be found in [12]. Modulo homotheties the three-dimensional cone \tilde{C} becomes the extended upper halfplane $\mathfrak{H}^* := \mathfrak{H} \cup \mathbb{Q} \cup \{\infty\}$, with $\partial \tilde{C}$ passing to the cusps $\mathfrak{H}^* \setminus \mathfrak{H}$. The 3-cones in \mathcal{V} tiling

² Although the Voronoi cones aren't necessarily simplicial, we can assume that they have been Γ -equivariantly subdivided.

C pass to the $\mathrm{SL}_2(\mathbb{Z})$ -translates of the ideal triangle with vertices at $0, 1, \infty$. Let us call these ideal triangles *Voronoi triangles*.

If $\xi \in \mathbf{C}_*^R$ is dual to a class in $H^1(\Gamma)$ and is supported on one 2-cone, then $\mathrm{supp} \xi$ passes to a geodesic μ between two cusps u_1, u_2 (Figure 2). We can subdivide μ into geodesic segments $\{\mu_i\}$ so that the endpoints e_i, e_{i+1} of μ_i lie in the same or adjacent Voronoi triangles. Then we assign cusps to the e_i as follows. If e_i is not an endpoint of ξ , then we assign any cusp c_i of the Voronoi triangle containing e_i . Otherwise, if $e_i = u_1$ or u_2 and hence is an endpoint of μ , then we assign e_i to itself. This determines a homology between ξ and a chain ξ'' supported on cones passing to the segments $[c_i, c_{i+1}]$. These cones are Voronoi cones, and thus $\xi'' \in \mathbf{C}_*^V$.

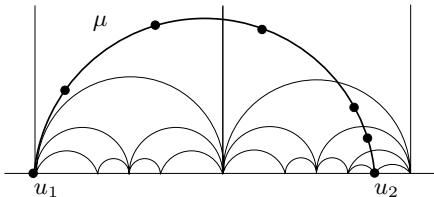


Fig. 2. A subdivision of μ ; the solid dots are the e_i . Since the e_i lie in the same or adjacent Voronoi triangles, we can assign cusps to them to construct a homology to a cycle in \mathbf{C}_*^V .

6 Well-Rounded Retracts

6.1

To conclude this article, we describe unpublished work of MacPherson and McConnell [16] that allows one to compute the Hecke action on those Γ for which a *well-rounded retract* W is available. Again for simplicity we focus on $\Gamma \subset \mathrm{SL}_n(\mathbb{Z})$; our first task is to explain what W is.

Let $V = \mathbb{R}^n$ with the standard inner product preserved by $\mathrm{SO}(n)$, and let $L \subset V$ be a lattice. For any $v \in V$, write $\|v\|$ for the length of v . Let $m(L)$ be the minimal nonzero length attained by any vector in L , and let $M(L) = \{v \in L \mid \|v\| = m(L)\}$. Then L is said to be *well-rounded* if $M(L)$ spans V .

6.2

Consider the space of cosets $Y = \mathrm{SL}_n(\mathbb{Z}) \backslash \mathrm{SL}_n(\mathbb{R})$. This space can be interpreted as the space of oriented lattices in \mathbb{R}^n modulo homotheties. Let $W \subset Y$ be the subset of well-rounded lattices, and for any $j = 0, \dots, n$, let $Y_j = \{L \in Y \mid \dim \mathrm{span} M(L) \geq j\}$. Clearly $Y_0 = Y$ and $Y_n = W$.

According to Ash [1], there is an $\mathrm{SO}(n)$ -equivariant retraction $r: Y \rightarrow W$ constructed as follows. Let $L \in Y_j$, and write $V = V_1 \oplus V_2$, where $V_1 = (\mathrm{span} M(L)) \otimes \mathbb{R}$, and V_2 is the orthogonal complement of V_1 . For $0 < \lambda \leq 1$, let $T(\lambda)$ be the linear transformation $(v_1, v_2) \mapsto (v_1, \lambda v_2)$, and let $L[\lambda]$ be the image of L under $T(\lambda)$. There is a critical value λ_0 for which $\dim \mathrm{span} M(L[\lambda]) > j$. Then we can define $r_j: Y_j \rightarrow Y_{j+1}$ by $r_j(L) = L[\lambda_0]$. These retractions can be composed to define the retraction $r: Y \rightarrow W$, and the space W is the well-rounded retract.

Since r is $\mathrm{SO}(n)$ -equivariant, it induces a retraction $\mathrm{SL}_n(\mathbb{Z}) \backslash \mathrm{SL}_n(\mathbb{R}) / \mathrm{SO}(n) \rightarrow W / \mathrm{SO}(n)$. Moreover, W can be given the structure of a locally-finite regular cell-complex. In a certain sense, these cells are dual to the Voronoï cones from §5: Voronoï cones of codimension k are in bijection with W -cells of dimension k . The construction works if Γ is replaced with any finite-index subgroup of $\mathrm{SL}_n(\mathbb{Z})$, and hence one has a convenient topological model to study the cohomology of any such Γ .

6.3

Now we consider how the ideas used in the construction of W can be applied to compute the action of the Hecke operators on cohomology. Let $d = (d_1, \dots, d_n)$ be a tuple of strictly positive integers, and let $g(d) \in \mathrm{GL}_n(\mathbb{Q})$ be the diagonal matrix with entries d . Let $\Gamma' := \Gamma \cap g^{-1} \Gamma g$. The *Hecke correspondence* associated to this data is the diagram $(c_1, c_2): \Gamma' \backslash X \rightarrow \Gamma \backslash X$, where the two maps are defined by $c_1(\Gamma'x) = \Gamma x$ and $c_2(\Gamma'x) = \Gamma gx$. In terms of the above description, $c_1^{-1} \circ c_2$ is the (multivalued) map that takes any lattice L to the set of sublattices $\{M \subset L \mid L/M \cong \mathbb{Z}/d_1\mathbb{Z} \oplus \dots \oplus \mathbb{Z}/d_n\mathbb{Z}\}$. A Hecke correspondence induces a map $c_1^* \circ (c_2)_*$ on cohomology that is exactly a classical Hecke operator. For example, if $n = 2$, p is a prime, and $d = (1, p)$, then the induced Hecke operator is the usual T_p .

6.4

Fix a tuple d and a pair of lattices $M \subset L$ as above. Choose $u \in [1, \infty)$. For $v \in L$, let $\| \|_u$ be $\|v\|$ if $v \in M$, and $u \cdot \|v\|$ otherwise. Now we can consider the retraction r described in §6.2, but using $\| \|_u$ instead of $\| \|$ as the notion of length. When $u = 1$, the result is the usual retract W . But for $u = u_0$ sufficiently large, only vectors in M will be detected in the retraction. Since M is itself a lattice, we have $W_{u_0} \cong W$.

These two complexes W_1 and W_{u_0} appear in a larger complex \mathcal{W} that depends on n and d and is fibered over the interval $[1, u_0]$ with fiber W_u . The fibers W_1 and W_{u_0} map to W by the maps c_1 and c_2 , respectively. One computes the action of the Hecke operator by lifting a class on $\Gamma \backslash W$ to $\Gamma' \backslash \mathcal{W}$, pushing the lift across $\Gamma' \backslash \mathcal{W}$ to the face $\Gamma \backslash W_{u_0}$, and then pushing down via c_2 to $\Gamma \backslash W$.

References

1. A. Ash, *Small-dimensional classifying spaces for arithmetic subgroups of general linear groups*, Duke Math. J. **51** (1984), 459–468.
2. ———, *Unstable cohomology of $SL(n, \mathcal{O})$* , J. Algebra **167** (1994), no. 2, 330–342.
3. A. Ash, D. Grayson, and P. Green, *Computations of cuspidal cohomology of congruence subgroups of $SL_3(\mathbb{Z})$* , J. Number Theory **19** (1984), 412–436.
4. A. Ash, P. E. Gunnells, and M. McConnell, *Cohomology of congruence subgroups of $SL_4(\mathbb{Z})$* , preprint, 2000.
5. A. Ash and M. McConnell, *Experimental indications of three-dimensional Galois representations from the cohomology of $SL(3, \mathbb{Z})$* , Experiment. Math. **1** (1992), no. 3, 209–223.
6. A. Ash, D. Mumford, M. Rapoport, and Y. Tai., *Smooth compactifications of locally symmetric varieties*, Math. Sci. Press, Brookline, Mass., 1975.
7. A. Ash, R. Pinch, and R. Taylor, *An \widehat{A}_4 extension of \mathbb{Q} attached to a non-selfdual automorphic form on $GL(3)$* , Math. Ann. **291** (1991), 753–766.
8. A. Ash and L. Rudolph, *The modular symbol and continued fractions in higher dimensions*, Invent. Math. **55** (1979), 241–250.
9. A. Borel and J.-P. Serre, *Corners and arithmetic groups*, Comm. Math. Helv. **48** (1973), 436–491.
10. J. Bygott, *Modular symbols and computation of cusp forms over imaginary quadratic fields*, Ph.D. thesis, Exeter University, 1997.
11. J. E. Cremona, *Hyperbolic tessellations, modular symbols, and elliptic curves over complex quadratic fields*, Compositio Math. **51** (1984), no. 3, 275–324.
12. P. E. Gunnells, *Modular symbols for \mathbb{Q} -rank one groups and Voronoï reduction*, J. Number Theory **75** (1999), no. 2, 198–219.
13. ———, *Computing Hecke eigenvalues below the cohomological dimension*, Experiment. Math. (to appear), 2000.
14. ———, *Symplectic modular symbols*, Duke Math. J., (to appear), 2000.
15. P. E. Gunnells and M. McConnell, *Hecke operators and \mathbb{Q} -groups associated to self-adjoint homogeneous cones*, math.NT/9811133, 1998.
16. R. MacPherson and M. McConnell, *Explicit reduction theory for Hecke correspondences*, in preparation.
17. Y.-I. Manin, *Parabolic points and zeta-functions of modular curves*, Math. USSR Izvestija **6** (1972), no. 1, 19–63.
18. L. Merel, *Universal Fourier expansions of modular forms*, On Artin's conjecture for odd 2-dimensional representations, Springer, Berlin, 1994, pp. 59–94.
19. J. Scholten, *A non-selfdual 4-dimensional Galois representation*, available from <http://www.math.uiuc.edu/Algebraic-Number-Theory/>, 1999.
20. J. T. Teitelbaum, *Modular symbols for $\mathbb{F}_q(T)$* , Duke Math. J. **68** (1992), no. 2, 271–295.
21. B. van Geemen and J. Top, *A non-selfdual automorphic representation of GL_3 and a Galois representation*, Invent. Math. **117** (1994), no. 3, 391–401.
22. B. van Geemen, W. van der Kallen, J. Top, and A. Verberkmoes, *Hecke eigenforms in the cohomology of congruence subgroups of $SL(3, \mathbb{Z})$* , Experiment. Math. **6** (1997), no. 2, 163–174.
23. G. Voronoï, *Sur quelques propriétés des formes quadratiques positives parfaites*, J. Reine Angew. Math. **133** (1908), 97–178.

Fast Jacobian Group Arithmetic on C_{ab} Curves

Ryuichi Harasawa and Joe Suzuki

Department of Mathematics, Graduate School of Science, Osaka University
1-1 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
`{harasawa,suzuki}@math.sci.osaka-u.ac.jp`

Abstract. The goal of this paper is to describe a practical and efficient algorithm for computing in the Jacobian of a large class of algebraic curves over a finite field. For elliptic and hyperelliptic curves, there exists an algorithm for performing Jacobian group arithmetic in $O(g^2)$ operations in the base field, where g is the genus of a curve. The main problem in this paper is whether there exists a method to perform the arithmetic in more general curves. Galbraith, Paulus, and Smart proposed an algorithm to complete the arithmetic in $O(g^2)$ operations in the base field for the so-called superelliptic curves. We generalize the algorithm to the class of C_{ab} curves, which includes superelliptic curves as a special case. Furthermore, in the case of C_{ab} curves, we show that the proposed algorithm is not just general but more efficient than the previous algorithm as a parameter a in C_{ab} curves grows large.

Keywords: Discrete logarithm problem, algebraic curve cryptography, Jacobian group, ideal class group, superelliptic curves, C_{ab} curves

1 Introduction

This paper is motivated by cryptography based on the intractability of the discrete logarithm problem (DLP) in the divisor class group of a curve. While elliptic curve cryptography has drawn considerable public attention in recent years, cryptosystems using hyperelliptic curves are currently getting accepted as well, which seems to be based on the following considerations:

1. the order of a Jacobian group can be large compared to the size of the field if the genus g of the curve is large (the Hasse-Weil bound [15]);
2. a novel method for solving the elliptic curve DLP that would be proposed in the future may not be applied to non-elliptic curves; and
3. recently, several fast algorithms for performing arithmetic on hyperelliptic curves have been proposed.

For elliptic curves, a method for performing addition among Jacobians has been known from a long ago, and its group arithmetic is given as a simple formula [13]. On the other hand, an efficient method of Jacobian group arithmetic for hyperelliptic curves has been given by D.G. Cantor [2]. (Although Cantor assumed the characteristic is not two, N. Koblitz recently excluded the constraint [7].) The only problem in addition of divisor classes is to compute good prescribed

representatives of a class. In the case of hyperelliptic curves, following Cantor [2] several methods for this have been proposed (see N. Smart [14] for details), and the algorithms realized in $O(g^2)$ operations in the base field are supposed to be the most efficient methods thus far.

In this paper, we address the problem whether or not there exists a method for performing Jacobian group arithmetic in $O(g^2)$ operations in the base field for more general curves than elliptic and hyperelliptic curves.

This problem has been solved in the affirmative for a class of curves called superelliptic curves (Galbraith, Paulus, and Smart [5]):

$$C/F_q : Y^a = \sum_{i=0}^b \alpha_i X^i ,$$

where $\alpha_i \in F_q$, $\alpha_b \neq 0$, a and b are coprime, and the curve is assumed to be nonsingular as an affine plane. In superelliptic curves, $a = 2$ implies a hyperelliptic curve, and $a = 2, b = 3$ implies an elliptic curve.

In this paper, we consider more general curves called C_{ab} curves [9]:

$$C/F_q : \sum_{\substack{0 \leq i \leq b, 0 \leq j \leq a, 0 \leq ai + bj \leq ab}} \alpha_{i,j} X^i Y^j = 0 ,$$

where $\alpha_{i,j} \in F_q$, $\alpha_{b,0} \neq 0$, $\alpha_{0,a} \neq 0$, and the curve is assumed to be nonsingular as an affine plane.

Previous methods for computing Jacobians [1,5] are based on the fact that a Jacobian group is isomorphic to the ideal class group of the coordinate ring $F_q[x, y]$ with $x = X \bmod C$ and $y = Y \bmod C$ in a canonical manner, which holds for C_{ab} curves. They reduce the problem of finding a good representative for a divisor class to that of finding a good representative of the corresponding ideal class (see Section 3).

On the other hand, Galbraith, S.Paulus, and Smart [5] reduced the problem of finding the representative element of each ideal class in a superelliptic curve to that of finding a minimal element in a lattice belonging to ideal in the ideal class, where the minimization is taken based on a certain metric suitable for superelliptic curves (see Section 4 for details), and applied an LLL-like algorithm [3] which ensures to find the minimal solution for this setting (S. Paulus [11]). In particular, in Paulus's LLL-like algorithm, division between polynomials is not required, so that Galbraith et. al's method [5] computes Jacobian group arithmetic in $O(g^2)$ operations in the base field (see Section 4 for details).

S. Arita [1] reduced the problem of finding the representative element of each ideal class for a C_{ab} curve to that of finding the minimal element in an ideal belonging to the ideal class, where the minimization is taken based on a certain monomial order suitable for C_{ab} curves (see Section 5 for details), and applied the so-called Buchberger algorithm that computes the reduced Gröbner basis. However, it is generally hard to evaluate the computational effort of finding a Gröbner basis of an ideal in a strict manner. Even in Arita's heuristic analysis, computing Jacobian group arithmetic is supposed to take $O(g^3)$ operations in the base field.

In this paper, we generalize Galbraith et. al's method [5] to C_{ab} curves, so that there does exist a method which performs Jacobian group arithmetic on C_{ab} curves in $O(g^2)$ operations in the base field. To this end, we first point out that the lattice reduction in Galbraith et. al. [5] is essentially equivalent to the problem of finding the minimal element in an ideal with respect to the C_{ab} order. We further modify Paulus's LLL-like algorithm for the lattice using a C_{ab} curve. As a result, we prove that the modification gives a more efficient algorithm. Moreover, we propose an efficient method for computing the inverse ideal of an ideal in the coordinate ring of a C_{ab} curve (see Section 6 for details). We will see that the method proposed in [5] for computing an inverse ideal is specific to the case of superelliptic curves. On the other hand, it turns out that a certain method for computing an inverse ideal in number fields works quite well for function fields defined using a C_{ab} curve.

2 Superelliptic and C_{ab} Curves

The notation follows [13] [15].

2.1 Superelliptic Curves

Definition 1 ([5]). A superelliptic curve defined over K is a nonsingular curve given as follows:

$$Y^a = \sum_{0 \leq i \leq b} \alpha_i X^i , \quad (1)$$

where $\alpha_i \in K$, $\alpha_{b,0} \neq 0$, a and b are coprime, and $\text{char}(K)$ does not divide a .

By definition, in elliptic and hyperelliptic curves we have $a = 2$, $b = 3$, and $a = 2$, $b \geq 3$, respectively. Then, the genus of a superelliptic curve is given [5] by

$$g = (a - 1)(b - 1)/2 . \quad (2)$$

2.2 C_{ab} Curves

Let C be a curve defined over K with at least one K -rational point P . Then, if we define $M_P := \{-v_P(f) | f \in L(\infty P)\}$, M_P makes a unitary semigroup under addition.

Definition 2 (C_{ab} Curve). If the semi-group M_P is generated by two positive integers a and b with $\text{g.c.d}(a, b) = 1$, the pair (C, P) is said a C_{ab} curve.

Let (C, P) be a C_{ab} curve. By definition, there exist functions $X \in L(\infty P)$ and $Y \in L(\infty P)$ with pole orders a and b at P , respectively. Using these two functions X and Y , we obtain the affine model of the C_{ab} curve as follows [9]:

$$C/K : \sum_{0 \leq i \leq b, 0 \leq j \leq a, ai + bj \leq ab} \alpha_{i,j} X^i Y^j = 0 , \quad (3)$$

where $\alpha_{i,j} \in K$, $\alpha_{b,0} \neq 0$, and $\alpha_{0,a} \neq 0$. The affine model in (3) is said the Miura canonical form of the C_{ab} curve (C, P) . In the Miura canonical form, a C_{ab} curve is assumed to be nonsingular in the affine plane, and P is the only infinite place P_∞ of curve C [9].

We assume that a C_{ab} curve is given in a Miura canonical form. Then, it turns out that C_{ab} curves include superelliptic curves with the same (a, b) . In fact, superelliptic curves are C_{ab} curves with $\alpha_{i,j} = 0$ ($0 \leq i \leq b$ and $1 \leq j \leq a - 1$) and $\alpha_{i,a} = 0$ ($1 \leq i \leq b$).

As for superelliptic curves, the formula

$$g = (a - 1)(b - 1)/2 \quad (4)$$

holds also for C_{ab} curves.

Definition 3 (C_{ab} Order [9]). *We order as $\alpha >_{ab} \beta$ for $\alpha = (\alpha_1, \alpha_2), \beta = (\beta_1, \beta_2) \in Z_{\geq 0}^2$ if one of the following two conditions holds:*

1. $a\alpha_1 + b\alpha_2 > a\beta_1 + b\beta_2$, or
2. $a\alpha_1 + b\alpha_2 = a\beta_1 + b\beta_2$, $\alpha_1 \leq \beta_1$.

By definition, under the condition $C(x, y) = 0$, monomials $X^{\alpha_1}Y^{\alpha_2}$ are ordered based on the pole order at infinity P_∞ :

$$-v_{P_\infty}(X^{\alpha_1}Y^{\alpha_2}) = a\alpha_1 + b\alpha_2 ,$$

and if they are equal, we suppose that the larger the degree with respect to X , the smaller the monomial order.

Similarly, polynomials $f = \sum \alpha_{i,j} X^i Y^j$ can be ordered according to the pole order at infinity P_∞ :

$$-v_{P_\infty}(f) = \max_{i,j, \alpha_{i,j} \neq 0} \{ai + bj\}.$$

3 Isomorphism between Jacobian and Ideal Class Groups

Jacobian group arithmetic on C_{ab} can be realized using the fact that the Jacobian group is isomorphic to the ideal class group of the coordinate ring for superelliptic and C_{ab} curves [1,5].

Definition 4. *If $D \in \text{Div}_K^0(C)$ is expressed as $E - nP_\infty$ with $E \geq 0$ and $P_\infty \notin \text{support}(E)$, D is said a semi-reduced divisor.*

Lemma 1 ([1,5]). *For each $j \in J_K(C)$, there exists a semi-reduced divisor $D \in \text{Div}_K^0(C)$ such that $j = [D]$.*

Definition 5. *If n is minimized in $D_1 = E - nP_\infty$ with $E \geq 0$ and $P_\infty \notin \text{support}(E)$ (semi-reduced) and $D_1 \sim D \in \text{Div}_K^0(C)$, then D_1 is said the reduced divisor equivalent to D .*

Lemma 2 ([1,5]). *If $D = E - nP_\infty \in \text{Div}_K^0(C)$ with $E \geq 0$ and $P_\infty \notin \text{support}(E)$ is a reduced divisor, then the reduced divisor $D_1 \sim D$ is unique for each $D \in \text{Div}_K^0(C)$, and $\deg(E) \leq g$*

We can obtain reduced divisors using the following algorithm [1,5]

Algorithm 1.

Input: Semi-reduced divisor $D = E - nP_\infty \in \text{Div}_K^0(C)$ with $E \geq 0$ and $P_\infty \notin \text{support}(E)$.

Output: The reduced divisor $G \sim -D$.

Step 1: Find $f \in L(\infty P_\infty)$ satisfying $(f)_0 \geq E$ and the pole order $-v_{P_\infty}(f)$ is minimal, where $L(\infty P_\infty) := \cup_{i=0}^{i=\infty} L(iP_\infty)$.

Step 2: $G \leftarrow -D + (f)$.

Since Algorithm 1 outputs a divisor equivalent to (-1) times the input divisor, if Algorithm 1 is applied twice, a divisor equivalent to the input divisor can be obtained.

However, directly dealing with divisors is not generally efficient because of irreducible decomposition of polynomials. So, Arita [1] and Galbraith et. al. [5] independently proposed Jacobian group arithmetic using ideal representation.

Since C_{ab} curve (C, P_∞) is nonsingular in the affine plane, the coordinate ring $K[x, y]$ with $C(x, y) = 0$ is a Dedekind domain. For a C_{ab} curve (C, P_∞) , an isomorphism Φ between the Jacobian group $J_K(C)$ and the ideal class group $H(K[x, y])$ of $K[x, y]$ is given as follows:

$$\Phi : J_K(C) \rightarrow H(K[x, y]) ,$$

$$[\sum_{P \in C, P \neq P_\infty} n_P P - (\sum_{P \in C, P \neq P_\infty} n_P) P_\infty] \mapsto [L(\infty P_\infty - \sum_{P \in C, P \neq P_\infty} n_P P)], \quad (5)$$

where we denote the ideal class which ideal $I \subset K[x, y]$ belongs to by $[I]$.

We call the ideals corresponding to reduced and semi-reduced divisors the reduced and semi-reduced ideals, respectively; then each semi-reduced ideal I is expressed by an integral ideal $I = L(\infty P_\infty - E) \subset L(\infty P_\infty) = K[x, y]$ with $E \geq 0$ and $P_\infty \notin \text{support}(E)$.

Now each integral ideal of $K[x, y]$ is a $K[x]$ -module, and if a $K[x]$ -basis is given as $(\beta_0, \dots, \beta_{a-1})$ with $\beta_i = \sum_{j=0}^{a-1} \beta_{i,j}(x)y^j$, the $K[x]$ -basis can be uniquely expressed by taking the Hermite normal form (HNF) of the matrix $(\beta_{i,j})$ (see Appendix). Therefore, we express each representative element of an ideal class group in $K[x, y]$ by the HNF of the $K[x]$ -basis.

Definition 6. We define the degree of a (fractional) ideal in $K[x, y]$ to be a degree of x in the product of the diagonal elements (subtracted by the degree of the denominator) of the HNF.

Then, it turns out that the degree of an ideal coincides with a value of n in the corresponding semi-reduced divisor $E - nP_\infty$. Hence, the sum of the degrees

with respect to x in each column of the HNF of a reduced ideal is at most g (see Lemma 2). It is known that the product of diagonal elements in the HNF expression of I is the norm of I [5].

Hence, Algorithm 1 can be replaced by

Algorithm 2.

Input: Semi-reduced ideal I .

Output: The reduced ideal $J \sim I^{-1}$.

Step 1: Find $f \in I$, $f \neq 0$ such that the pole order $-v_{P_\infty}(f)$ is minimal.

Step 2: $J \leftarrow (f)I^{-1}$.

4 Jacobian Group Arithmetic on Superelliptic Curves

Galbraith et. al. [5] proposed an algorithm (Algorithm 3) for performing Jacobian group arithmetic on superelliptic curves. Algorithm 3 below computes a $K[x]$ -basis to represent an ideal in an ideal class: we embed $K[x, y]$ into $(K[x])^a$ with

$$\phi : K[x, y] \rightarrow (K[x])^a$$

$$\sum_{0 \leq i \leq a-1} c_i(x)y^i \mapsto (c_0(x), \dots, c_{a-1}(x))$$

and define the metric of $C = (c_0(x), \dots, c_{a-1}(x)) \in (K[x])^a$ as follows: $|C| := \max|C|_i$ where $|C|_i := \deg_x(c_i(x)) + \frac{b}{a}i$. Consider an ideal $I \subset K[x, y]$ and let $\{f_0, \dots, f_{a-1}\}$ be a $K[x]$ -basis of I ; then, $\phi(I)$ is a lattice generated by $\{\phi(f_i)\}_i$ over $K[x]$, so that minimization over $f \in I$ with respect to $-v_{P_\infty}(f)$ is equivalent to minimization over $u \in \phi(I)$ with respect to $|u|$ ($-v_{P_\infty}(f) = a|\phi(f)|$ for $f \in I$).

Galbraith et. al. [5] apply Paulus's method [11] in the following way.

Definition 7 ([5]). The orthogonality defect $OD(f_0, \dots, f_{a-1})$ of a basis $\{f_0, \dots, f_{a-1}\}$ for a lattice L is defined as

$$OD(f_0, \dots, f_{a-1}) := \sum_i |f_i| - \deg_x(d(L)),$$

where $d(L) := \det(f_0^*, \dots, f_{a-1}^*)$ with $f_i^* := (f_0^i(x), f_1^i(x)x^{\frac{b}{a}}, \dots, f_{a-1}^i(x)x^{\frac{b}{a}(a-1)})^t$ for $f_i = \sum_{j=0}^{a-1} f_j^i(x)y^j$.

It is easy to see that $OD(f_0, \dots, f_{a-1}) \geq 0$.

Definition 8 ([11]). The basis $\{f_0, \dots, f_{a-1}\}$ for a lattice is said a reduced basis if $OD(f_0, \dots, f_{a-1}) = 0$.

Proposition 1 ([11]). Let $\{f_0, \dots, f_{a-1}\}$ be the reduced basis for an lattice L . Then $f \in \{f_0, \dots, f_{a-1}\}$ such that $|f| = \min_i\{|f_i|\}$ is the minimal nonzero element in L with respect to $|\cdot|$.

Algorithm 3 (Jacobian Group Arithmetic on Superelliptic Curves [5]).

Input: Reduced ideals I_1, I_2 in $K[x, y]$ (HNF).

Output: The reduced ideal $I_3 \sim I_1 I_2$ (HNF).

Step 1: $D \leftarrow I_1 I_2$;

Step 2: $J \leftarrow$ a semi-reduced ideal equivalent to D^{-1} ;

Step 3: $f \leftarrow$ a minimal nonzero element in J with respect to $|\phi(\cdot)|$.

Step 4: $I_3 \leftarrow$ the HNF of $(f) J^{-1}$.

The validity of Algorithm 3 can be easily checked: basically, the process of Algorithm 2 is done twice in Steps 2-4. (Note that J in Step 2 is not required to be a reduced ideal but I_3 in Step 4 is.) We now discuss some of these steps in detail; this will show that Algorithm 3 really uses superelliptic curves.

In Step 2, for $D = I_1 I_2$ an integral ideal equivalent to D^{-1} is computed using the formula

$$D^{-1} \sim \prod_{\sigma \in \text{Gal}(K(x, y)/K(x)), \sigma \neq 1} D^\sigma.$$

Note that here it is assumed that K contains the a -th roots of unity. So if necessary, the base field is extended in this step. Any $\sigma \in \text{Gal}(K(x, y)/K(x))$ is given by $y^\sigma = \rho y$ for some a -th root of unity ρ . Hence the conjugates D^σ and therefore also D^{-1} are easy to compute. It seems unclear how to extend this idea to more general C_{ab} curves.

For Step 3, we can obtain the minimal element by finding the reduced basis. The complexity of finding a reduced basis is given as follows:

Proposition 2 ([11]). *We can find the reduced basis from a $K[x]$ -basis $\{C_0, \dots, C_{a-1}\}$ of the lattice in*

$$O(a^3 \max|C_i| \times OD(C_0, \dots, C_{a-1}) \log^2 q). \quad (6)$$

For Step 4, since

$$I_3 = J^{-1}(f) = \frac{D}{\prod D^\sigma}(f) = \frac{I_1 I_2}{N_{K(x, y)/K(x)}(I_1 I_2)}(f),$$

and since the norm $N_{K(x, y)/K(x)}(I_1 I_2)$ is obtained computing the product of the diagonal elements in the HNF of the ideal $I_1 I_2$, the ideal I_3 can be easily computed [5].

In summary, the whole computation can be evaluated as in Proposition 3.

Proposition 3 ([5]). *Let C/K be a superelliptic curve. Jacobian group arithmetic on $\text{Jac}_K(C)$ (Algorithm 3) can be performed in $O(a^7 g^2 \log^2 q)$ if $a|q - 1$ and in $O(a^9 g^2 \log^2 q)$ if $a \nmid q - 1$*

5 Jacobian Group Arithmetic on C_{ab} Curves

Arita[1] proposed an algorithm (Algorithm 4) for performing Jacobian group arithmetic on C_{ab} curves. Algorithm 4 below computes a $K[x, y]$ -basis to represent a unique ideal in an ideal class. The idea is that in C_{ab} order, monomials are arranged according to the pole orders at infinity P_∞ when they are regarded as functions on a C_{ab} curve.

Algorithm 4 (Jacobian Group Arithmetic on C_{ab} Curves [1]).

Input: Reduced ideals I_1, I_2 in $K[x, y]$.

Output: The reduced ideal I_3 equivalent to ideal product I_1I_2 .

Step 1: $J \leftarrow I_1I_2$;

Step 2: $f \leftarrow$ the minimal nonzero element in J with respect to C_{ab} order;

Step 3: $h \leftarrow$ the minimal nonzero element with respect to C_{ab} order satisfying
 $(h)J \subseteq (f)$;

Step 4: $I_3 \leftarrow (h/f)J$.

The validity of Algorithm 4 can be easily checked: basically, the process of Algorithm 2 is done in Steps 2-4. (In particular, h and $(f)J^{-1}$ play the roles of the f and I in the second round of Algorithm 2, respectively.)

In Algorithm 4, the minimal element in an ideal is computed by finding the reduced Gröbner basis. (Note that a reduced Gröbner basis gives the unique representation of an ideal.) However, it takes much time to obtain a Gröbner basis, and it is hard to evaluate its computational effort in a strict manner. In [1], the computation of Step 2 is heuristically analyzed to be $O(g^3 \log^2 q)$ if the value of a is bounded.

However, to authors' knowledge, it seems that there has been none thus far to address Jacobian group arithmetic on C_{ab} curves except Algorithm 4 [1].

6 Fast Jacobian Group Arithmetic on C_{ab} Curves

From the considerations in the previous sections, it turns out that the following two problems should be solved for extending Galbraith et. al.'s method to C_{ab} curves:

1. how to compute the inverse ideal I^{-1} given an ideal I ; and
2. how to compute the minimal element over an ideal with respect to C_{ab} order.

6.1 Computing Inverse Ideals

For the first problem, we propose a more general method to obtain an inverse ideal than that in the case of superelliptic curves. The idea is based on the method for computing inverse ideals in the integral closure of a number field [3]. Let L be a number field, and Z_L the integral closure of L , and $n := [L : Q]$. We first fix the Z -basis $(w_i)_{1 \leq i \leq n}$ of Z_L .

Definition 9. *The different of L is defined as*

$$\Gamma(L) := \{x \in L \mid \text{Trace}_{L/Q}(xZ_L) \subset Z\}^{-1}. \quad (7)$$

Then, the following proposition follows [3]:

Proposition 4. *Let $(\omega_i)_{1 \leq i \leq n}$ be a Z -basis of Z_L and I an ideal of Z_L given by a matrix M whose columns give the coordinates of a Z -basis $(\gamma_i)_{1 \leq i \leq n}$ of I on the chosen Z -basis. Let $T = (t_{i,j})$ be the $n \times n$ matrix such that $t_{i,j} = \text{Trace}_{L/Q}(\omega_i \omega_j)$. Then, the columns of the matrix $(M^t T)^{-1}$ form a Z -basis of the ideal $(I\Gamma(L))^{-1}$.*

Therefore, for a given ideal $I \subset Z_L$, the ideal product $I\Gamma(L)^{-1}$ is computed by taking the HNF of the $n \times n^2$ matrix obtained from M and T^{-1} . If the HNF is N , then, by Proposition 4, $(N^t T)^{-1}$ forms a Z -basis of $(I\Gamma(L)^{-1})^{-1}\Gamma(L)^{-1} = I^{-1}$.

Now we go back to the case of C_{ab} curves. The ring $L(\infty P_\infty)$ is a Dedekind domain. Furthermore, since C_{ab} curves are generally nonsingular, $L(\infty P_\infty)$ coincides with the coordinate ring $K[x, y]$. Therefore, the integral closure of $K[x]$ in $K(x, y)$ is $K[x, y]$, so that the result for Z_L can be extended to $K[x, y]$ in a natural manner. Then, $1, y, \dots, y^{a-1}$ can be the $K[x]$ -basis of $K[x, y]$, and $T = (t_{i,j})_{1 \leq i \leq a, 1 \leq j \leq a}$ are given by $t_{i,j} = \text{Trace}_{K(x,y)/K(x)}(y^{i+j-2})$. The value of each $t_{i,j}$ can be computed using the Newton formula (page 163, [3]) if the definition equation is given. Let $D_i(x)$ and $C_l^{(i)}$ as $y^a = \sum_{i=0}^{a-1} D_i(x)y^i$ (the definition equation of a C_{ab} curve) and $y^i = \sum_{l=0}^{a-1} C_l^{(i)}(x)y^l$ ($a \leq i \leq 2a-2$), respectively, in $K[x, y]$ with $x = X \bmod C$ and $y = Y \bmod C$. Then, $\text{Trace}_{K(x,y)/K(x)}(1) = a$, $\text{Trace}_{K(x,y)/K(x)}(y) = D_{a-1}(x)$, for $i = 2, \dots, a-1$

$$\text{Trace}_{K(x,y)/K(x)}(y^i) = iD_{a-i}(x) + \sum_{l=1}^{i-1} D_{a-l}(x)\text{Trace}_{K(x,y)/K(x)}(y^{i-l}), \quad (8)$$

and for $i = a, \dots, 2a-2$

$$\text{Trace}_{K(x,y)/K(x)}(y^i) = \sum_{l=0}^{a-1} C_l^{(i)}(x)\text{Trace}_{K(x,y)/K(x)}(y^l). \quad (9)$$

If we compute and store the matrix dT^{-1} with d the determinant of T beforehand, we obtain:

Algorithm 5 (Computation of Inverse Ideals for C_{ab} Curves).

Input: Semi-reduced ideal I in $K[x, y]$ with $(\gamma_i)_{1 \leq i \leq a}$ a $K[x]$ -basis of I (HNF).
Output: The inverse ideal I^{-1} .

Step 1: $N \leftarrow$ the HNF of the $a \times a^2$ matrix $(\gamma_i \delta_j)$, with δ_j column vectors of dT^{-1} ;

Step 2: $h \leftarrow \det(N^t)$;

$$P \leftarrow dh(N^t T)^{-1} = (dT^{-1})(h(N^t)^{-1});$$

$$k \leftarrow \text{GCM}(\text{GCM}(P), h);$$

$$e \leftarrow \frac{h}{k};$$

$$W \leftarrow \frac{1}{k} P;$$

$$I^{-1} \leftarrow (W, e) \quad (I^{-1} = W(e)^{-1}).$$

($\text{GCM}(A)$ with A a matrix and $\text{GCM}(f, g)$ with $f, g \in K[x]$ denote the GCM of all the elements in A and that of f and g , respectively.)

Theorem 1. *Algorithm 5 is computed in $O(a^8 g^2 \log^2 q)$ and in $O(a^4 g^2 \log^2 q)$ for C_{ab} and superelliptic curves, respectively, if the degree of an ideal I is $O(g)$.*

Theorem 1 is obtained based on the following facts: if the degree of x in the determinant of an $m \times n$ matrix M is bounded by t ,

1. the Hermite normal form (HNF) of M with $\text{rank}(M) = m$ is obtained in $O(m^2 nt^2 \log^2 q)$ (for the proof, see Appendix);
2. if $n = m$, the determinant of M is obtained in $O(\max\{m^3 t \log^2 q, t^2 \log^2 q\})$ (for the proof, see Appendix);
3. if $n = m$, the inverse of M is obtained in $O(\max\{m^5 t \log^2 q, m^2 t^2 \log^2 q\})$. (computing the m^2 determinants yields the inverse ideal if Cramer's formula is applied);

and if the degrees of x in two polynomials f, g is bounded by s ,

4. the GCM of f and g is obtained in $O(s^2 \log^2 q)$.

Proof of Theorem 1:

1) General case

For Step 1, the degree of x in $\text{Trace}_{K(x,y)/K(x)}(y^i)$, $0 \leq i \leq a-1$, is $O(g)$: in fact, from $\deg_x[D_{a-l}(x)] \leq b$, $0 \leq l \leq a-1$, and (8), we have

$$\begin{aligned} \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^i)] &\leq \max_{1 \leq l \leq i} \{\deg_x[D_{a-l}(x)] + \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^{i-l})]\} \\ &\leq \max_{1 \leq l \leq i} \{b + \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^{i-l})]\} \\ &\leq ib + \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^0)] \\ &= ib. \end{aligned}$$

For $a \leq i \leq 2a-2$, one checks that the degree of x in $C_l^{(i)}(x)$ is at most $b(i-a+1)$ (In fact, $\deg_x(y^a) = b$, and $\deg_x(y^i) \leq b(i-a+1)$ implies $\deg_x(y^{i+1}) \leq b(i-a+1) + b$ for $i = a+1, \dots, 2a-2$), so that

$$\begin{aligned} \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^i)] &\leq \max_{0 \leq l \leq a-1} \{\deg_x[C_l^{(i)}(x)] + \deg_x[\text{Trace}_{K(x,y)/K(x)}(y^l)]\} \\ &\leq \max_{0 \leq l \leq a-1} \{b(i-a+1) + lb\} \\ &\leq ib, \end{aligned}$$

where (9) has been applied.

In any case, the degree of x in each element of T is $ib = O(g)$ (see (4)). If we apply Cramer's formula, the degree of x of each element in dT^{-1} with $d = \det(T)$ is bounded by $O(ag)$ since the degree of x of each element in T is at most g , so is $\deg_x(\delta_j)$.

On the other hand, by assumption the degree of x in each element in the HNF expressing the input ideal is at most g , i.e., $\deg_x(\gamma_j) \leq g$. Since there are a^2 pairs of $(\gamma_i \delta_j)_{i,j}$, they are obtained in $O(a^4 g^2 \log^2 q)$. Using 1 with $m = a$, $n = a^2$, and $t = O(a^2 g)$, the HNF N of the $a \times a^2$ matrix is obtained in $O(a^8 g^2 \log^2 q)$.

For Step 2, if we apply Cramer's formula, $(N^t)^{-1}$ and $\det(h)$ are computed in $O(a^7 g^2 \log^2 q)$ (use 3 and 2 with $m = a$ and $t = O(a^2 g)$, respectively). Since the degrees of x of each element in matrices dT^{-1} and $h(N^t)^{-1}$ are $O(ag)$ and $O(a^2 g)$ (note that the degree of x in each element of an HNF is at most a times as that of the original matrix), the degree of x in each element of matrix P is $O(a^2 g)$. Since the GCM of two polynomials of degree $O(a^2 g)$ is computed in $O(a^4 g^2 \log^2 q)$ (use 4 with $s = O(a^2 g)$), $GCM(GCM(P), h)$ is computed in $O(a^6 g^2 \log^2 q)$.

Since a^2 divisions between polynomials of degree $O(a^2 g)$ are done (recall that the degree of x in each element of P is $O(a^2 g)$, so is the degree of x in k), W is obtained in $O(a^6 g^2 \log^2 q)$.

Hence, Step 2 takes $O(a^6 g^2 \log^2 q)$.

Therefore, Algorithm 5 takes $O(a^8 g^2 \log^2 q)$.

2) the case of superelliptic curves

Let $y^a = f(x)$ be a C_{ab} curve with $\deg_x f(x) = b$.

For Step 1, the HNF representation of the ideal dT^{-1} is $[f(x), y, \dots, y^{a-1}]$. In fact, one checks

$$T = \begin{bmatrix} a & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & af(x) \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & af(x) & \cdots & 0 \\ 0 & af(x) & 0 & \cdots & 0 \end{bmatrix}, \quad dT^{-1} = \begin{bmatrix} f(x) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

and $d = a^a (f(x))^{a-1}$. Thus, the degree of the ideal expressed by dT^{-1} is $\deg_x f(x) = b$ since the HNF of dT^{-1} is

$$\begin{bmatrix} f(x) & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & \\ 0 & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

and the total degree of the diagonal elements is $\deg_x f(x) = b$. On the other hand, by assumption, the degree of the ideal expressed by (γ_i) is $O(g)$. So, the degree of the ideal expressed by the HNF N is $O(g) + b = O(g)$. Since $(\gamma_i \delta_j)_{i,j}$

are obtained in $O(a^2g^2 \log^2 q)$, from **1** with $m = a$, $n = a^2$, and $t = O(g)$, the HNF N is obtained in $O(a^4g^2 \log^2 q)$. And, the degree of the ideal N is $O(g+b) = O(g)$.

For Step 2, if we apply Cramer's formula, $(N^t)^{-1}$ and $\det(h)$ are computed in $O(\max\{a^5g \log^2 q, a^2g^2 \log^2 q\}) = O(a^4g^2 \log^2 q)$ (use **3** and **2** with $m = a$ and $t = O(g)$, respectively, and note $a = O(g)$). Since the degrees of x of each element in matrices dT^{-1} and $h(N^t)^{-1}$ are $O(g)$, the degree of x in each element of matrix P is $O(g)$. Since the GCM of two polynomials of degree $O(g)$ is computed in $O(g^2 \log^2 q)$ (use **4** with $s = O(g)$), $GCM(GCM(P), h)$ is computed in $O(a^2g^2 \log^2 q)$.

Since a^2 divisions between polynomials of degree $O(g)$ are done (recall that the degree of x in each element of P is $O(g)$, so is the degree of x in k), W is obtained in $O(a^4g^2 \log^2 q)$.

Hence, Step 2 takes $O(a^4g^2 \log^2 q)$.

Therefore, Algorithm 5 takes $O(a^4g^2 \log^2 q)$. \square

Note that in the proof of Theorem 1, degree of x in each element of W is bounded by $O(a^2g)$ and $O(g)$ for C_{ab} and superelliptic curves, respectively, which will be referred later.

6.2 Computing the Minimal Element

For the second problem, by the definition of the metric $|\cdot|$ in $(K[x])^a$, for $f = \sum_{i=0}^{a-1} f_i(x)y^i \in K[x, y]$ with $f_i(x) \in K[x]$, we have

$$-v_{P_\infty}(f) = \max_i \{a\deg_x(f_i(x)) + bi\} = a|\phi(f)|.$$

Therefore, for an ideal $I \subset K[x, y]$, minimization over I with respect to C_{ab} order is equivalent to minimization over $\phi(I)$ with respect to $|\cdot|$, so that for the second problem we can apply Paulus's method [11] (finding the reduced basis) to C_{ab} curves.

Proposition 5 ([11]). *Let b_1, \dots, b_n be a basis for a lattice L and denote by $b_{i,j}$ the j -th coordinate of b_i . If the coordinates of the vectors b_1, \dots, b_n can be permuted in such a way that they satisfy*

1. $|b_i| \leq |b_j|$ for $1 \leq i < j \leq n$; and
2. $|b_{i,j}| < |b_{i,i}| \geq |b_{i,k}|$ for $1 \leq j < i < k \leq n$.

Then b_1, \dots, b_n forms a reduced basis.

Now we go back to the case of C_{ab} curves. For a $K[x]$ -basis f_0, \dots, f_{a-1} for a lattice L , if it satisfies that $|f_i| - |f_j| \notin \mathbb{Z}$ ($0 \leq i < j \leq a-1$), then f_0, \dots, f_{a-1} forms a reduced basis by Proposition 5. (Note that $\text{g.c.d}(a, b) = 1$ implies there exists an unique l such that $|f| = |f|_l$ for a nonzero vector $f = (f_0, \dots, f_{a-1}) \in (K[x])^a$. In fact, if $|f| = |f|_i = |f|_j$ with $0 \leq i \leq j \leq a-1$, i.e. $ac_i + bi = ac_j + bj$, where c_i and c_j are the degrees of x in $f_i(x)$ and $f_j(x)$, respectively,

then $a(c_i - c_j) = b(j - i)$. Hence, $a|j - i$ since $\text{g.c.d}(a, b) = 1$, which implies $i = j$.)

Therefore, we can modify Paulus's algorithm [11] to obtain the following algorithm.

Algorithm 6. (Computation of Reduced Basis in C_{ab} Curves)

Input: $K[x]$ -basis $\{f_0, \dots, f_{a-1}\}$ for a lattice L with $f_i = (f_{i,0}(x), \dots, f_{i,a-1}(x))^t$.

Output: The reduced basis.

Step 1: $g_0 \leftarrow f_0$, $k \leftarrow 1$;

Step 2: $g_k \leftarrow f_k$;

Step 3: if $|g_j| - |g_k| \notin Z$ ($\forall j < k$) then $k \leftarrow k + 1$, otherwise go to Step 5-1;

Step 4 if $k = a$ then output $\{g_0, \dots, g_{a-1}\}$, otherwise go to Step 2;

Step 5-1 let j, l be the indices such that $|g_j| - |g_k| \in Z$, $|g_j| = |g_j|_l$, $|g_k| = |g_k|_l$;

Step 5-2 if $|g_j| > |g_k|$ then swap g_j and g_k ;

Step 5-3 $g_k \leftarrow g_k - rx^{|g_k|-|g_j|}g_j$ with $r = c_{k,l}/c_{j,l}$, where $c_{k,l}$ and $c_{j,l}$ are the leading coefficients of $g_{k,l}(x)$ and $g_{j,l}(x)$, respectively; and

Step 6 if $\sum_{j=0}^k |g_j| + \sum_{j=k+1}^{a-1} |f_j| = \deg_x d(L)$ then output $\{g_0, \dots, g_k, f_{k+1}, \dots, f_{a-1}\}$, otherwise go to Step 3.

The validity of Algorithm 6 can be checked by Definition 8 and Proposition 5.

Theorem 2. Algorithm 6 is computed in $O(a^3 t(t+b) \log^2 q)$ if the degree of x in $(f_{i,j})_{i,j}$ is bounded by t .

Proof of Theorem 2:

It is easy to check that Step 5-3 dominates the computational complexity of Algorithm 6. In Step 5-3, the computation of $g_k \leftarrow g_k - rx^{|g_k|-|g_j|}g_j$ requires shift operations and $O(at)$ multiplications in K . Note that $OD(g_0, \dots, g_k, f_{k+1}, \dots, f_{a-1})$ strictly decreases after executing Step 5-3. Therefore, the number of iterations of executing Step 5-3 is bounded by $a \times (OD(f_0, \dots, f_{a-1}) - \deg_x d(L)) \leq a \times OD(f_0, \dots, f_{a-1}) = O(a(\sum_{i=0}^{a-1} (t+b))) = O(a^2(t+b))$. Hence, Algorithm 6 is computed in $O(at \log^2 q \times O(a^2(t+b))) = O(a^3 t(t+b) \log^2 q) \square$

For Steps 3 and 5, in [5], Paulus's original algorithm was directly applied in a straightforward manner that a set of linear equations

$$\sum_{j=0}^{k-1} c_{j,i}^* r_j = c_{k,i}^* \quad (0 \leq i \leq k-1)$$

is solved for r_j , $j = 0, \dots, k-1$, every time k and $OD(f_0, \dots, f_{a-1})$ are updated, where $c_{j,i}^*$ is the leading coefficient of $g_{j,i}(x)$ at the order of the leading term of in g_j with respect to C_{ab} order (if no such a coefficient exists in $g_{j,i}(x)$, then $c_{j,i}^* = 0$) and $c_{j,i}^* = 0$ for $0 \leq j < i \leq k-1$ (if necessary, swap rows in each g_j), so that the leading term in g_k can be cancelled out with either of g_0, \dots, g_{k-1} .

They estimated the complexity of solving the equations as $O(k^2)$ operations in the base field since the coefficient matrix $(c_{j,i}^*)_{0 \leq i,j \leq k-1}$ is a lower triangular matrix (thus, that of computing a reduced basis as $O(a^7 g^2 \log^2 q)$), which we considered too large for implementation. In this paper, we find that the solution is quite simple, i.e. we only solve one linear equation (see Step 5-3) since all r_j except one are equal to zeros, which is completed in $O(1)$ operations in the base field. In fact, we have

1. for each column in the coefficient matrix $(c_{j,i})_{0 \leq i,j \leq k-1}$ and the column vector $(c_{k,i})_{0 \leq i \leq k-1}$, all the elements except one are equal to zeros; and
2. for each row in the coefficient matrix $(c_{j,i})_{0 \leq i,j \leq k-1}$, all the elements except one are equal to zeros.

(Apparently, $c_{j,i} = c_{j,i}^* \Leftrightarrow c_{j,i}^* \neq 0 \Leftrightarrow |g_j| = |g_{j,i}|$), where the first property is from the assumption $\text{g.c.d}(a, b) = 1$, and the second from Step 3 in Algorithm 6, i.e. $|g_j| - |g_i| \notin Z$ ($0 \leq i < j < k-1$).

Algorithm 6 utilizes these properties, so that the computational effort has been greatly saved.

In summary, we see that the extension of Galbraith et. al.'s method to C_{ab} curves is possible. The whole proposed algorithm can be described as in Algorithm 7.

Algorithm 7. (proposed Jacobian group arithmetic on C_{ab} curves)

Input: Ideals I_1, I_2 in $K[x, y]$ (HNF).

Output: The reduced ideal I_3 equivalent to $I_1 I_2$ (HNF).

Step 1: $J \leftarrow$ the HNF of $I_1 I_2$;

Step 2: Applying Algorithm 5 to J , $J^{-1} \leftarrow (W, e)$;

Step 3: Applying Algorithm 6 to W , $f \leftarrow$ the minimal element in W with respect to C_{ab} order;

Step 4: $I_3 \leftarrow$ the HNF of $(f)W^{-1} = (f/e)J$.

Our final task is to ensure that Algorithm 7 is completed in $O(g^2)$ operations in the base field if the sizes of a and q are bounded, which is the goal of this paper. The computation time of Steps 1 and 4 is basically the same as those in Algorithm 3, which is within $O(g^2)$ operations in the base field. Step 2 is completed in $O(a^8 g^2 \log^2 q)$ by Theorem 1 since the degree of ideal J is $O(g)$. For Step 3, since the degree of x in each element of the matrix W is $O(a^2 g)$, Step 3 is completed in $O(a^3 \cdot a^2 g \cdot (a^2 g + b) \cdot \log^2 q) = O(a^7 g^2 \log^2 q)$ by Theorem 2. Hence we obtain:

Theorem 3. Algorithm 7 is completed in $O(a^8 g^2 \log^2 q)$ and in $O(a^4 g^2 \log^2 q)$ for C_{ab} and superelliptic curves, respectively.
(See Table 1 for details.)

Table 1. Complexity of Jacobian Group Arithmetic

	Proposed method		Galbraith, Paulus and Smart [5]
	C_{ab}	superelliptic	superelliptic
Step 1 (ideal product)	$O(a^4 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$
Step 2 (inverse ideal)	$O(a^8 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$	$O(a^7 g^2 \log^2 q)$ $(O(a^9 g^2 \log^2 q))$
Step 3 (minimal element)	$O(a^7 g^2 \log^2 q)$ (substitute $t = a^2 g$ to Theorem 2)	$O(a^3 g^2 \log^2 q)$ (substitute $t = g$ to Theorem 2)	$O(a^7 g^2 \log^2 q)$ (apply Proposition 2)
Step 4 (ideal product)	$O(a^7 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$
whole process	$O(a^8 g^2 \log^2 q)$	$O(a^4 g^2 \log^2 q)$	$O(a^7 g^2 \log^2 q)$ $(O(a^9 g^2 \log^2 q))$

7 Concluding Remarks

We proposed a fast Jacobian group arithmetic algorithm for C_{ab} curves (Algorithm 7), evaluated the complexity of the proposed algorithm. As a result, it turned out that Algorithm 7 is more efficient than Algorithms 3 (Galbraith et. al.) in the case of superelliptic curves (Proposition 3, Theorem 3). Furthermore, although Algorithm 7 can be applied to C_{ab} curves as well as superelliptic curves, Algorithm 7 completes the arithmetic in $O(g^2)$ operations in the base field while Algorithm 4 does in $O(g^3)$ operations in the base field.

Future work includes exploring a faster Jacobian group arithmetic scheme for more general curves.

Acknowledgements

The authors would like to thank Shinji Miura and Junji Shikata for their useful comments.

References

1. Seigo Arita, *Algorithms for Computations in Jacobian Group of C_{ab} Curve and Their Application to Discrete-Log Based Public Key Cryptosystems*, IEICE Trans part A Vol. J82-A No.8, 1291-1299, Aug. 1999. in Japanese.
2. D. G. Cantor, *Computing in the Jacobian of a hyper-elliptic curves*, Math.Comp., **48** (1987), pp. 95-101.
3. H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, GTM 138, 1993.

4. G. Frey and H. Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation **62** (1994), 865-874.
5. S. D. Galbraith, S. Paulus, and N. P. Smart, *Arithmetic on Superelliptic Curves*, preprint, 1998.
6. R. Hartshorne, *Algebraic Geometry*, Springer-Verlag, GTM 52, 1977.
7. N. Koblitz, *Hyperelliptic cryptosystems*, J. Cryptography, Vol. 1, 139-150, 1989.
8. V. S. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptography CRYPTO '85 (Lecture Notes in Computer Science, vol 218), Springer-Verlag, 1986, pp. 417-426.
9. Shinji Miura, *The study of error coding codes based on algebraic geometry*, Dr. thesis. in Japanese (1997).
10. Achim Müller, *Effiziente Algorithmen für Probleme der linearen Algebra über Z* , Master's thesis, Universität des Saarlandes, Saarbrücken, 1994.
11. S. Paulus, *Lattice basis reduction in function field* in Ants-3, Algorithmic Number Theory(Lecture Notes in Computer Science, vol 1423), 567-575, 1998.
12. S. Paulus and A. Stein, *Comparing Real and Imaginary Arithmetics for Divisor Class Groups of Hyperelliptic Curves* in Ants-3, Algorithmic Number Theory(Lecture Notes in Computer Science, vol 1423), 576-591, 1998.
13. J. H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Math., vol. 106, Springer-Verlag, Berlin and New York, 1994.
14. N. P. Smart, *On the performance of Hyperelliptic Cryptosystems*, Advances in Cryptology EUROCRYPTO'99 (Lecture Notes in Computer Science vol 1592), 165-175, 1998.
15. H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer Universitext, Springer-Verlag, 1993.

Appendix : Hermite Normal Form (HNF) with $K[x]$ Coefficients

Definition 10. We say that an $m \times n$ matrix $A = (a_{i,j})$ with $K[x]$ coefficients is an Hermite normal form (HNF) if there exists $r \leq n$ and a strictly increasing map f from $[r+1, n]$ to $[1, m]$ satisfying the following properties:

1. for $r+1 \leq j \leq n$, $a_{f(j),j} \neq 0$, $a_{i,j} = 0$ if $i > f(j)$; and for $k < j$
 - (a) $\deg_x(a_{f(k),j}) < \deg_x(a_{f(k),k})$ if $\deg_x(a_{f(k),k}) \geq 1$; or
 - (b) $a_{f(k),j} = 0$ if $\deg_x(a_{f(k),k}) = 0$ (equivalently, $a_{f(k),k} \in K$)
2. the first r columns of A are equal to 0.
3. $a_{f(k),k}$, $k = r+1, \dots, n$, are monic.

Proposition 6. Let $A = (a_{i,j})$ be an $m \times n$ matrix with $K[x]$ coefficients. Then, there exists a unique $m \times n$ matrix B in HNF of the form $B = AU$ with $U \in \text{GL}_n(K[x])$, where $\text{GL}_n(K[x])$ is the group of matrices with $K[x]$ coefficients which are invertible, i.e. whose determinant belongs to K .

We call the matrix consisting of the last $n - r$ columns the HNF of A .

When we compute an HNF directly, it is hard to evaluate its complexity since we don't know how large the degree of x grows during the process. But,

in the case of integer coefficients and $\text{rank}(A) = m$, if we know the value D that is a multiple of the determinant of the Z -module $L(A)$ generated by the columns of A , then we can compute the HNF of A by using D [3]. And this modified method requires $O(m^2n|D|^2)$ -bit operations, where $|D|$ is the number of bits for expressing D . (Note that in the case of a finite field, the computation of an HNF takes $O(m^2n)$ operations in the field [3].) Therefore we obtain the following algorithm by extending the result for Z to $K[x]$ in a natural manner.

Algorithm 8. HNF

Input: $m \times n$ matrix A with $K[x]$ coefficients and $\text{rank}(A) = m$.

Output: The HNF of A .

Step 1: the $R \leftarrow$ the $m \times m$ matrix whose columns consist of linear independent column vectors of A ;

Step 2: $D \leftarrow \det(R)$;

Step 3: Compute the HNF modulus D [3];

Remark 1. 1. In the case of $m = n$, Step 1 is not required.

2. In Step 2, since $L(R)$ is an sub-module of $L(A)$, the value of D is a multiple of $\det(L(A))$, where $(L(A))$ is a $K[x]$ -module generated by the columns of A .

Proposition 7. We assume the degree of x in the determinant of A is less than t . If $q > t$, then Algorithm 8 is completed in $O(m^2nt^2 \log^2 q)$.

Remark 2. We consider the case where g is extraordinarily large, say $q = 2^{160}$ (common in cryptography etc.), so that the condition $q > t$ is always cleared. Otherwise, no computational problem arises.

Proof:

For Step 1, let a_1, \dots, a_n be the column vectors which A consists of, and $A_i = [a_1, \dots, a_i]$ be the matrix that consists of the first i columns of A . We consider $W \subset K$ of cardinality t (such a W always exists because $\#(W) = t < q = \#(K)$). Then, we have

$$\deg_x(\det(L(A))) < t = \deg_x(\prod_{\alpha \in W} f_\alpha). \quad (10)$$

Let $r_\alpha(i) := \text{rank}_{K[x]/(f_\alpha(x))}(A_i \bmod f_\alpha(x))$. Then, we can show that there exists an $f_\alpha(x)$ such that $\text{rank}(A) = r_\alpha(n)$. Suppose $\text{rank}(A) < r_\alpha(n)$ for all $\alpha \in W$. Then, $\det(L(A)) \bmod f_\alpha = 0$ for all $\alpha \in W$. But, this implies $\prod_{\alpha \in W} f_\alpha$ divides $\det(L(A))$, which contradics (10).

So, we can construct linear independent column vectors of A , i.e. Step 1 can be broken down into the following stages:

Stage 1 choose an $f_\alpha \in W$, and for each $1 \leq i \leq n$ compute $r_\alpha(i)$;

Stage 2 if there exists an l such that $r_\alpha(l) = m$, go to Stage 4-1;

Stage 3 $W \leftarrow W - \{f_\alpha\}$ and go to Stage 1;

Stage 4-1 if $r_\alpha(1) = 1$, then choose a_1 , otherwise throw away a_1 ; and

Stage 4-2 for each $2 \leq i \leq l$, choose a_i such that $r_\alpha(i-1) < r(i)$.

It is clear that the computation of Stage 1 dominates Step 1. We can obtain the value of $r_\alpha(i)$, $1 \leq i \leq n$, by computing the HNF of the $a \times a^2$ matrix $A \bmod f_\alpha(x)$. From $K[x]/(f_\alpha(x)) \cong K$ and the fact that the number of iterations in Stage 1 is bounded by $\#(W)$, it turns out that Step 1 takes $\#(W) \times O(m^2n \log^2 q) = O(t) \times O(m^2n \log^2 q) = O(m^2nt \log^2 q)$ (the HNF is obtained in $O(m^2n \log^2 q)$ if each element of the element is in K [3], which is much smaller than that for $K[x]$).

For Step 2, we can obtain the value of D by computing $D \bmod f_\alpha(x)$ for each $\alpha \in W$ and applying the Chinese Remainde Theorem. It takes $\#(W) \times O(m^3 \log^2 q) = O(m^3t \log^2 q)$ to compute $D \bmod f_\alpha(x)$ for all $\alpha \in W$. Then, $D = \sum_\alpha g_\alpha(D \bmod f_\alpha(x))$, where $g_\alpha = s_\alpha h_\alpha$ with $r_\alpha f_\alpha + s_\alpha h_\alpha = 1$, where $r_\alpha \in K[x]$ and $h_\alpha = \prod_{\alpha' \in W} f_{\alpha'}/f_\alpha$. The multiplication $\prod_{\alpha \in W} f_\alpha$ is done in $O(\sum_{i=1}^{\#(W)} i \cdot \log^2 q) = O(t^2 \log^2 q)$; the division between $\prod_{\alpha \in W} f_\alpha$ and f_α is done in $O(t \cdot \log^2 q)$ since the degrees of x in the two polynomials are $t-1$ and 1 ; s_α is computed in $O(1 \cdot \log^2 q)$ (Proposition 3 [12]); the multiplication $s_\alpha h_\alpha$ is done in $O(t \log^2 q)$ since the degrees of x in s_α and h_α are 0 and $t-1$; and the final computation $\sum_\alpha g_\alpha(D \bmod f_\alpha(x))$ takes $\#(W) \times O(1 \cdot (t-1) \log^2 q) = t \times O(t \log^2 q)$ since the degrees of x in h_α is $t-1$ and $D \bmod f_\alpha(x) \in K$. Hence, Step 2 takes $O(\max\{m^3t \log^2 q, t^2 \log^2 q\})$.

Step 3 takes $O(m^2n(t \log q)^2) = O(m^2nt^2 \log^2 q)$ [3], since the number of bits expressing D is $O(t \log q)$,

Since $m \leq n$, Algorithm 8 is completed in $O(m^2nt^2 \log^2 q)$. \square

Lifting Elliptic Curves and Solving the Elliptic Curve Discrete Logarithm Problem

Ming-Deh A. Huang, Ka Lam Kueh, and Ki-Seng Tan

¹ Department of Computer Science, University of Southern California
Los Angeles, CA 90089-0781
huang@pollux.usc.edu

² Institute of Mathematics, Academia Sinica, Taipei, Taiwan
maklk@ccvax.sinica.edu.tw

³ Department of Mathematics, National Taiwan University, Taipei, Taiwan
tan@math.ntu.edu.tw

Abstract. Essentially all subexponential time algorithms for the discrete logarithm problem over finite fields are based on the index calculus idea. In proposing cryptosystems based on the elliptic curve discrete logarithm problem (ECDLP) Miller [6] also gave heuristic reasoning as to why the index calculus idea may not extend to solve the analogous problem on elliptic curves. A careful analysis by Silverman and Suzuki provides strong theoretical and numerical evidence in support of Miller's arguments. An alternative approach recently proposed by Silverman, dubbed 'xedni calculus', for attacking the ECDLP was also shown unlikely to work asymptotically by Silverman himself and others in a subsequent analysis. The results in this paper strengthen the observations of Miller, Silverman and others by deriving necessary but difficult-to-satisfy conditions for index-calculus type of methods to solve the ECDLP in subexponential time. Our analysis highlights the fundamental obstruction as being the necessity to lift an asymptotically increasing number of random points on an elliptic curve over a finite field to rational points of reasonably bounded height on an elliptic curve over \mathbb{Q} . This difficulty is underscored by the fact that a method that meets the requirement implies, by virtue of a theorem we prove, a method for constructing elliptic curves over \mathbb{Q} of arbitrarily large rank.

1 Introduction

In the elliptic curve discrete logarithm problem (ECDLP), we are given an elliptic curve E over a finite field \mathbb{F}_q and two points P and Q on the curve, and the problem is to find an integer n (if it exists) such that $Q = nP$. The ECDLP is an analog of the discrete logarithm problem over finite fields, which is the basis of many public key cryptosystems. Miller [6] and Koblitz [3] independently proposed public key cryptosystems based on the elliptic curve discrete logarithm problem. In proposing such cryptosystems Miller [6] also gave heuristic reasoning as to why the index calculus idea, which lies at the heart of all the subexponential algorithms for the discrete logarithm problem, may not extend to solve the elliptic curve discrete logarithm problem.

The classical index calculus method for the discrete logarithm problem works by lifting the problem from a finite field to the ring of integers, where there is much richer arithmetic structure to take advantage of. To extend this idea to work for the ECDLP, it is natural to consider lifting an elliptic curve E/\mathbb{F}_p of interest to some elliptic curve \mathcal{E}/\mathbb{Q} in order to possibly take advantage of the structure of $\mathcal{E}(\mathbb{Q})$. Miller pointed out the difficulty for such an approach is at least two fold: first in lifting the curve E to a curve \mathcal{E} of sufficiently large rank over \mathbb{Q} , then in actually lifting points from E to rational points of reasonably bounded height on \mathcal{E} . A careful analysis by Silverman and Suzuki in [10] provides strong theoretical and numerical evidence in support of Miller's arguments.

Silverman [8] proposed an alternative approach, dubbed the ‘xedni calculus’, for attacking the ECDLP. The xedni idea ‘turns the index calculus on its head’ by first lifting a bounded number (nine) of points to \mathbb{Q} then finding a lift \mathcal{E}/\mathbb{Q} of E to fit the lifted points. This approach circumvents the difficulty of lifting points and does not require the lift \mathcal{E} for E to have a large rank. In fact the success of this method depends on the lifted points being linearly dependent in $\mathcal{E}(\mathbb{Q})$. The probability for this to occur would presumably be low. To increase the probability Silverman imposed additional conditions on the lift based on some heuristic arguments involving the Birch-Swinnerton-Dyer Conjecture. However, a subsequent analysis by Silverman and Jacobson et al [2] shows that with the xedni algorithm the probability of success in finding a discrete logarithm on an elliptic curve over a finite field is in fact negligible asymptotically speaking.

The results in this paper strengthen the observations of Miller [6] and the analysis of Silverman et al [2,10] by deriving necessary but difficult-to-satisfy conditions for any index-calculus type of method which involves the lifting idea to solve the ECDLP in subexponential time.

The center piece of our analysis is the following result concerning lifting an elliptic curve over a finite fields together with a finite set of points. Let E be an elliptic curve over a finite field \mathbb{F}_p . For $r \in \mathbb{Z}_{>0}$ and $h \in \mathbb{R}_{>0}$, let $n_E(r, h)$ denote the number of $(r+1)$ -tuples $\lambda = (P_0, \dots, P_r)$ with P_i in some cyclic subgroup of $E(\mathbb{F}_p)$ so that (E, λ) can be lifted to some $(\mathcal{E}_\lambda, \Lambda)$ over \mathbb{Q} with the rank of $\mathcal{E}(\mathbb{Q})$ bounded by r and the canonical heights of the points in Λ bounded by h . We show that $n_E(r, h)$ is bounded by $2^{O(r^3)}(h/\log |\Delta|)^{O(r^2)}N^r$ where $N = |E(\mathbb{F}_p)|$.

From the theorem we deduce the following conclusions.

With the approach such as the index calculus method, where one lifts an elliptic curve E/\mathbb{F}_p to an elliptic curve \mathcal{E}/\mathbb{Q} before lifting random points (generated from the two points in question), in order to possibly achieve subexponential running time (such as $O(\exp(c(\log p)^{1/2}(\log \log p)^{1/2}))$, the rank of \mathcal{E} needs to grow at least as fast as $(\log p)^{1/4}$ as p grows.

With the approach such as the xedni calculus method, where one lifts a set of random points (generated from the two points in question) then constructs a curve \mathcal{E} to fit the lifted points, in order to possibly achieve subexponential running time (such as $O(\exp(c(\log p)^{1/2}(\log \log p)^{1/2}))$, the number of lifted points needs to grow at least as fast as $(\log p)^{1/4}$ as p grows. To underscore the difficulty in meeting this condition, we show that a method for lifting an asymptotically

increasing number (such as $(\log p)^{1/4}$) of random points on an elliptic curve over \mathbb{F}_p to rational points of canonical height bounded subexponential in $\log p$ on an elliptic curve over \mathbb{Q} implies a method for constructing elliptic curves over \mathbb{Q} of arbitrarily large rank. On the other hand, bounding the number of lifted points, as the xedni algorithm in [8], results in asymptotically negligible probability of success in solving the ECDLP.

Our analysis depends on a conjecture of Lang [4] that the canonical height of any nonzero rational point on an elliptic curve \mathcal{E} over \mathbb{Q} is bounded from below by $c \log |\Delta(\mathcal{E})|$ where c is a universal constant independent of \mathcal{E} and $\Delta(\mathcal{E})$ is the minimal discriminant of \mathcal{E} . Lang's conjecture is the only unproven assumption needed throughout this paper. (The results in [2,10] depend on Lang's conjecture as well as other heuristic assumptions.) It is worth mentioning that the conjecture has been proven to a large extent [1,9].

It should be pointed out that our results are asymptotic in nature and they leave open the possibility for the index-calculus idea (including the xedni method) to successfully attack the ECDLP in the lower range of p .

The rest of this paper is organized as follows. In Section 2 we prove the theorem concerning the lifting problem and in Section 3 we relate the result to the elliptic curve discrete logarithm problem.

2 The Lifting Problem

Let E be an elliptic curve over a finite field \mathbb{F}_p and $\lambda = (P_1, \dots, P_m)$ with $P_i \in E(\mathbb{F}_p)$. Let \mathcal{E} be an elliptic curve over \mathbb{Q} and $\Lambda = (\mathcal{P}_1, \dots, \mathcal{P}_m)$ with $\mathcal{P}_i \in \mathcal{E}(\mathbb{Q})$. We say that (E, λ) is lifted to (\mathcal{E}, Λ) if E can be obtained as the reduction of \mathcal{E} modulo p with P_i as the reduction of \mathcal{P}_i modulo p for $i = 1, \dots, m$. We say that λ is lifted with E with canonical height bounded by h if the canonical height of \mathcal{P}_i is bounded by h for $i = 1, \dots, m$.

Let $\hat{h}(\mathcal{P})$ denote the canonical height [7] of \mathcal{P} for $\mathcal{P} \in \mathcal{E}(\mathbb{Q})$. Let

$$N(\mathcal{E}, b) = \#\{\mathcal{P} \in \mathcal{E}(\mathbb{Q}) : \hat{h}(\mathcal{P}) \leq b\}.$$

Let $r = r(\mathcal{E})$ be the rank of $\mathcal{E}(\mathbb{Q})$, T be the number of torsion points in $\mathcal{E}(\mathbb{Q})$, and R be the regulator of \mathcal{E} over \mathbb{Q} . Then it is known [4] that

$$N(\mathcal{E}, b) \approx T \alpha_r \left(\frac{b}{R^{1/r}} \right)^{r/2},$$

where α_r is the volume of the unit r -ball. We assume Lang's conjecture [4] that

$$\hat{h}(\mathcal{P}) \geq c \log |\Delta(\mathcal{E})|$$

for some constant c independent of \mathcal{E} , where $\Delta(\mathcal{E})$ denotes the minimal discriminant of \mathcal{E} . Then from

$$R^{1/r} \geq \left(\frac{\sqrt{3}}{2} \right)^{r-1} \min \hat{h}(\mathcal{P})$$

where the minimum is over all nonzero $\mathcal{P} \in \mathcal{E}(\mathbb{Q})$ (see [4]), and

$$\alpha_r \approx \frac{1}{\sqrt{\pi r}} \left(\frac{2\pi e}{r} \right)^{r/2},$$

and that $T \leq 16$ (see [5]), it follows that for $r \geq 1$

$$N(\mathcal{E}, b) \leq 2^{c_1 r^2} \left(\frac{b}{\log |\Delta|} \right)^{r/2} \quad (1)$$

for some positive constant c_1 independent of \mathcal{E} .

Proposition 1. *There exists a positive constant c such that for all elliptic curves \mathcal{E} defined over \mathbb{Q} , if the rank of $\mathcal{E}(\mathbb{Q})$ is no greater than r , then for any $\mathcal{P}_0, \dots, \mathcal{P}_r$ in $\mathcal{E}(\mathbb{Q})$ with $h(\mathcal{P}_i) \leq h$, there exist integers c_i with $|c_i| \leq 2^{cr^2} (\frac{h}{\log |\Delta|})^{r/2}$ such that $\sum_i c_i \mathcal{P}_i = 0$, where Δ is the minimal discriminant of \mathcal{E} .*

Proof: For $\mathcal{P} \in \mathcal{E}(\mathbb{Q})$, let $\|\mathcal{P}\| = \sqrt{\hat{h}(\mathcal{P})}$. For $a_i \in \{0, \dots, m-1\}$,

$$\left\| \sum_{i=0}^r a_i \mathcal{P}_i \right\| \leq \sum_{i=0}^r |a_i| \|\mathcal{P}_i\| \leq \sqrt{h} \sum_{i=0}^r |a_i| \leq m(r+1)\sqrt{h}.$$

So

$$\hat{h} \left(\sum_{i=0}^r a_i \mathcal{P}_i \right) \leq m^2(r+1)^2 h.$$

Since the number of (a_0, \dots, a_r) with $a_i \in \{0, \dots, m-1\}$ is m^{r+1} , if

$$N(\mathcal{E}, m^2(r+1)^2 h) < m^{r+1}, \quad (2)$$

then there must exist two distinct (a_0, \dots, a_r) and (b_0, \dots, b_r) with $a_i, b_i \in \{0, \dots, m-1\}$ such that $\sum_i a_i \mathcal{P}_i = \sum_i b_i \mathcal{P}_i$, and hence $\sum_i c_i \mathcal{P}_i = 0$ with $c_i = a_i - b_i$, so $|c_i| < m$. From Eq. (1),

$$N(\mathcal{E}, m^2(r+1)^2 h) < 2^{c_1 r^2} \left(\frac{m^2(r+1)^2 h}{\log |\Delta|} \right)^{r/2}$$

for some constant c_1 independent of \mathcal{E} . Hence (2) holds if $m > 2^{cr^2} (h/\log |\Delta|)^{r/2}$ where c is a constant independent of \mathcal{E} .

Theorem 1. *Let E be an elliptic curve over a finite field \mathbb{F}_p . For $r \in \mathbb{Z}_{>0}$ and $h \in \mathbb{R}_{>0}$, let $n_E(r, h)$ denote the number of $\lambda = (P_0, \dots, P_r)$ with P_i in some cyclic subgroup of $E(\mathbb{F}_p)$ so that (E, λ) can be lifted to some (\mathcal{E}, Λ) over \mathbb{Q} with the canonical heights of the points in Λ bounded by h and the rank of $\mathcal{E}(\mathbb{Q})$ bounded by r . Then $n_E(r, h)$ is bounded by $2^{O(r^3)} (h/\log |\Delta|)^{O(r^2)} N^r$ where $N = |E(\mathbb{F}_p)|$ and Δ is the minimal discriminant of \mathcal{E} .*

Proof: Let $\lambda = (P_0, \dots, P_r)$ with P_i in some cyclic subgroup of $E(\mathbb{F}_p)$ with a generator S . Suppose (E, λ) is lifted to some (\mathcal{E}, Λ) with canonical height bounded by h . Suppose $\Lambda = (\mathcal{P}_0, \dots, \mathcal{P}_r)$. If the rank of $\mathcal{E}(\mathbb{Q})$ is bounded by r , then from Proposition 1 it follows that there exist integers c_i such that

$$\sum_i c_i \mathcal{P}_i = 0,$$

where

$$|c_i| \leq 2^{cr^2} \left(\frac{h}{\log |\Delta|} \right)^{r/2}$$

and Δ is the minimal discriminant of \mathcal{E} .

Suppose $P_i = m_i S$. Then

$$0 = \sum_i c_i P_i = \left(\sum_i c_i m_i \right) S.$$

So

$$\sum_i c_i m_i \equiv 0 \pmod{N}$$

where N is the order of S . Now $n_E(r, h)$ is bounded by the number of (m_0, \dots, m_r) such that $\sum_i c_i m_i \equiv 0 \pmod{N}$ and $|c_i|$ is bounded by

$$M = 2^{cr^2} \left(\frac{h}{\log |\Delta|} \right)^{r/2}.$$

For each $c = (c_0, \dots, c_r)$, let n_c denote the number of $(m_0, \dots, m_r) \pmod{N}$ such that

$$c_0 m_0 + \dots + c_r m_r \equiv 0 \pmod{N}.$$

Suppose the g.c.d. of c_0, \dots, c_r is g , then

$$n_c \leq gN^r \leq MN^r.$$

So

$$n_E(r, h) \leq (2M + 1)^{r+1} MN^r = 2^{O(r^3)} (h/\log |\Delta|)^{O(r^2)} N^r.$$

3 Analysis on the Index-Calculus Approach to ECDLP

In the elliptic curve discrete logarithm problem we are given an elliptic curve E over a finite field \mathbb{F}_p , and two points $S, T \in E(\mathbb{F}_p)$, and the problem is to find an integer m (if it exists) so that $mS = T$.

A natural generalization of the index calculus method for the ECDLP can be outlined as follows.

1. Find an elliptic curve \mathcal{E} defined over \mathbb{Q} whose reduction mod p is E . Suppose $\mathcal{E}(\mathbb{Q})$ has rank r with a basis \mathcal{P}_i , $i = 1, \dots, r$, and suppose $P_i \in E(\mathbb{F}_p)$ is the reduction of \mathcal{P}_i mod p .

2. For random integer j , lift jS to some $S' \in \mathcal{E}(\mathbb{Q})$, and write S' in terms of \mathcal{P}_i (up to a torsion point). Each S' yields a linear relation on the discrete logarithms of P_i . With r many linearly independent relations we can solve for the discrete logarithms $\log_S(P_i)$ of all P_i .
3. For random integer j , lift $T + jS$ to some $S' \in \mathcal{E}(\mathbb{Q})$, and write S' in terms of \mathcal{P}_i . Then $\log_S(T)$ can be determined.

For the method to work in subexponential time, $r + 1$ random points in $E(\mathbb{F}_p)$ need to be lifted to points in $\mathcal{E}(\mathbb{Q})$ of canonical height bounded by some h which can be at most subexponential in $\log p$. The number of such $(r + 1)$ -tuples of points in $E(\mathbb{F}_p)$ cannot be greater than $n_E(r, h)$, which by Theorem 1 is bounded by $2^{O(r^3)}(h/\log|\Delta|)^{O(r^2)}N^r$. Hence the success probability is bounded by $\frac{2^{O(r^3)}(h/\log|\Delta|)^{O(r^2)}}{N}$. Since N can be in the order of p , for the success probability to be at least $1/\exp[(\log p)^{1/2}(\log \log p)^{1/2}]$, say, it is necessary that $r^2 \log h > c' \log p$ for some constant c' . Since h can be at most $\exp[O(1)(\log p)^{1/2}(\log \log p)^{1/2}]$, the number of lifted points $r + 1$ needs to be at least in the order of $(\log p)^{1/4}$ as p grows.

The same conclusion can also be deduced from Proposition 1. Let h be an upper bound on $\hat{h}(\mathcal{P}_i)$ and $\hat{h}(S')$ where S' lifts a point P in $E(\mathbb{F}_p)$. Then from Proposition 1 it follows that there exist integers c_i with absolute values bounded by $2^{cr^2}(h/\log|\Delta|)^{r/2}$ such that

$$c_0 S' + c_1 \mathcal{P}_1 + \dots + c_r \mathcal{P}_r = 0$$

so

$$c_0 P + c_1 P_1 + \dots + c_r P_r = 0.$$

The number of $P \in E(\mathbb{F}_p)$ satisfying

$$c_0 P + c_1 P_1 + \dots + c_r P_r = 0$$

with $|c_i| \leq 2^{cr^2}(h/\log|\Delta|)^{r/2}$ is bounded by $2^{O(r^3)}h^{O(r^2)}$.

It follows that the probability that a random P can be lifted to some S' with height bounded by h is no greater than $\frac{2^{O(r^3)}h^{O(r^2)}}{p}$. For the probability to be at least $1/\exp[(\log p)^{1/2}(\log \log p)^{1/2}]$, say, it is necessary that $r^2 \log h > c' \log p$ for some constant c' . Even if we allow the points to be lifted to subexponential canonical height so that h is about $\exp[(\log p)^{1/2}(\log \log p)^{1/2}]$, the rank r of \mathcal{E} still needs to be at least in the order of $(\log p)^{1/4}$.

Note that the observation above holds regardless of the method used to construct \mathcal{E} and lift a point from E to \mathcal{E} . The fact that the rank of \mathcal{E} needs to grow at least as fast as $(\log p)^{1/4}$ as p grows already poses a significant difficulty for the index calculus method to work.

Next we turn our attention to the xedni calculus method for the elliptic curve discrete logarithm problem. Below is a general outline for the method.

1. Generate random P_0, \dots, P_r with $P_i = a_iS + b_iT$ where a_i, b_i are random integers.
2. Lift P_i to some \mathcal{P}_i over \mathbb{Q} , then construct an elliptic curve \mathcal{E} over \mathbb{Q} so that the pair \mathcal{E} and $(\mathcal{P}_0, \dots, \mathcal{P}_r)$ is a lift of E and (P_0, \dots, P_r) .
3. If the rank of $\mathcal{E}(\mathbb{Q})$ is no greater than r , then $\mathcal{P}_0, \dots, \mathcal{P}_r$ are integrally dependent, so that

$$\sum_i c_i \mathcal{P}_i = 0$$

for some integers c_i , then upon reduction mod p we have

$$0 = \sum_i c_i (a_i S + b_i T) = (\sum_i c_i a_i) S + (\sum_i c_i b_i) T.$$

From this the discrete logarithm of T in terms of S can be obtained with high probability, since a_i and b_i are randomly chosen.

The xedni algorithm of Silverman is consistent with the outline above, with r set at 9, and as mentioned before, additional conditions imposed on \mathcal{E} .

For the method to work in subexponential time, $r+1$ random points in $E(\mathbb{F}_p)$ need to be lifted to points of canonical height at most subexponential in $\log p$ on some \mathcal{E} over \mathbb{Q} of rank at most r . The number of random $(r+1)$ -tuples $\lambda = (P_0, \dots, P_r)$ is bounded by N^{r+1} . For a λ to lead to a success in finding the discrete logarithm we need (E, λ) to be lifted to some (\mathcal{E}, Λ) over \mathbb{Q} with the canonical heights of the points in Λ bounded by h and the rank of $\mathcal{E}(\mathbb{Q})$ bounded by r . The number of such $(r+1)$ -tuples cannot be greater than $n_E(r, h)$, which by Theorem 1 is bounded by $2^{O(r^3)}(h/\log |\Delta|)^{O(r^2)}N^r$. Hence the success probability is bounded by $\frac{2^{O(r^3)}(h/\log |\Delta|)^{O(r^2)}}{N}$. Since N can be in the order of p , for the success probability to be at least $1/\exp[(\log p)^{1/2}(\log \log p)^{1/2}]$, say, it is necessary that $r^2 \log h > c' \log p$ for some constant c' . Since h is at most $\exp[O(1)(\log p)^{1/2}(\log \log p)^{1/2}]$, the number of lifted points $r+1$ needs to be at least in the order of $(\log p)^{1/4}$ as p grows. This is true regardless of how the curve \mathcal{E} is constructed for each $(r+1)$ -tuple of points in $E(\mathbb{F}_p)$. In particular, for bounded r (such as the case with the xedni method in [8]), the probability of success tends to zero asymptotically with p . Hence the xedni calculus method as described in [8] cannot work as a subexponential algorithm asymptotically.

To extend the scope of applicability of the xedni calculus idea, we would need to increase the the number of random points on an elliptic curve to be lifted to rational points of reasonably bounded canonical height on an elliptic curve over \mathbb{Q} . But the difficulty of such task is underscored by that of constructing elliptic curves of large rank over \mathbb{Q} as reasoned below.

Let $m_E(r, h)$ denote the number of $\lambda = (P_0, \dots, P_r)$ with P_i in $E(\mathbb{F}_p)$ so that (E, λ) can be lifted to some (\mathcal{E}, Λ) over \mathbb{Q} with the canonical heights of the points in Λ bounded by h . For any fixed r , suppose for elliptic curves E over \mathbb{F}_p , $m_E(r, h) \geq N^{r+1}/p^c$ where where c is a positive constant less than 1 and $\log h/\log p$ tends to 0 as p tends to infinity (say h is subexponential in $\log p$). Then by Theorem 1, $m_E(r, h) > n_E(r, h)$ for sufficiently large p and E with

cyclic group $E(\mathbb{F}_p)$. It follows that for sufficiently large p , some elliptic curve over \mathbb{Q} lifting some elliptic curve E over \mathbb{F}_p together with some r -tuple of points on E must have rank at least r .

Acknowledgement

We would like to thank Joe Silverman for reading an earlier draft of this paper and for his valuable suggestions.

References

1. M. Hindry and J. Silverman, *The canonical height and integral points on elliptic curves*, Invent. Math., 93, (1988), 419-450.
2. M.J. Jacobson, N. Koblitz, J.H. Silverman, A. Stein, and E. Teske, Analysis of the Xedni Calculus Attack, Preprint.
3. N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.*, 48, pp. 203-209, 1987.
4. S. Lang, *Fundamental of Diophantine Geometry*, Springer-Verlag, 1983.
5. B. Mazur, Modular curves and Eisenstein ideal, *I.H.E.S. Publ. Math.* 47 (1977), 33-186.
6. V. Miller, The use of elliptic curves in cryptography. *Advances in Cryptography*, Ed. H.C. Williams, Springer-Verlag, 1986, 417-426.
7. J.H. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, 1986.
8. J.H. Silverman, The xedni calculus and the elliptic curve discrete logarithm problem, preprint.
9. J.H. Silverman, *Lower bound for the canonical height on elliptic curves*, Duke Math. J., 48 (1981), 633-648.
10. J.H. Silverman and J. Suzuki, Elliptic curve discrete logarithms and the index calculus, *Advances in Cryptology -Asiacrypt '98*, Springer-Verlag, 1998, 110-125.

A One Round Protocol for Tripartite Diffie–Hellman

Antoine Joux

SCSSI, 18, rue du Dr. Zamenhoff
F-92131 Issy-les-Mx Cedex, France
Antoine.Joux@ens.fr

Abstract. In this paper, we propose a three participants variation of the Diffie–Hellman protocol. This variation is based on the Weil and Tate pairings on elliptic curves, which were first used in cryptography as cryptanalytic tools for reducing the discrete logarithm problem on some elliptic curves to the discrete logarithm problem in a finite field.

1 Introduction

Since its discovery in 1976, the Diffie–Hellman protocol has become one of the most famous and largely used cryptographic primitive. In its basic version, it is an efficient solution to the problem of creating a common secret between two participants. Since this protocol is also used as a building block in many complex cryptographic protocols, finding a generalization of Diffie–Hellman would give a new tool and might lead to new and more efficient protocols.

In this paper, we show that the Weil and Tate pairings can be used to build a tripartite generalization of the Diffie–Hellman protocol. These pairings were first used in cryptography as cryptanalytic tools to reduce the complexity of the discrete logarithm problem on some “weak” elliptic curves. Of course, the problem of setting a common key between more than two participants has already been addressed (see the protocol for conference keying in [1]). However, all the known techniques require at least two round of communication. In some protocols having these two rounds can be somewhat cumbersome, and a single round would be much preferable. To give an example, exchanging an email message key with a two round Diffie–Hellman protocol would require both participants to be connected at the same time, which is a very undesirable property for a key exchange protocol. For this reason, we believe that the one round tripartite Diffie–Hellman presented here is a real improvement over conference keying even though the computational cost will be somewhat higher.

2 The Discrete Logarithm Problem on Weak Elliptic Curve

The discrete logarithm problem on elliptic curves is now playing an increasingly important role in cryptography. When elliptic curve cryptosystems where first

proposed in [9], computing the number of points of a given curve was a challenging task, since the Schoof, Elkies and Atkin algorithm was not yet mature (for a survey of this algorithm see [6]). For this reason and also to simplify the addition formulas, the idea of using special curves quickly arose. However, it was shown later on that some of these special cases are not good enough. Today, three weak special cases have been identified. In one of them, the discrete logarithm problem becomes easy (i.e. polynomial time) as was shown in [11,10]. This easiest case happens when the number of points of the elliptic curve over \mathbb{F}_p is exactly p . In the two other cases, the discrete logarithm problem on the elliptic curve is transformed into a discrete logarithm problem in a small extension of the field of definition of the elliptic curve. These two reductions are called the Menezes, Okamoto, Vanstone (MOV) reduction [8] and the Frey, Rück (FR) reduction [3]. A survey of these reductions was published at Eurocrypt'99 [4], and gave a comparison of these two reductions. The conclusion was the FR reduction can be applied to more curves than the MOV reduction and moreover that it can be computed faster than the MOV reduction. Thus for all practical usage, the authors recommend the FR reduction. However, they claim that the computation of the FR and MOV reduction may be a heavy load. We will show that in fact this is not the case and that these reductions can be turned from cryptanalytic to cryptographic tools.

Pairings on Elliptic Curve

The MOV and FR reductions are both based on a bilinear pairing, in the MOV case it is the Weil pairing and in the FR case it is (a variation of) the Tate pairing. In the sequel, we describe these pairings for an elliptic curve E defined over \mathbb{F}_p . In order to define these pairings, we first need to introduce the function field and the divisors of the elliptic curve. Very informally, the function field $K(E)$ of E is the set of rational map in x and y modulo the equation of E (e.g. $y^2 - x^3 - ax - b$). A divisor D is an element of the free group generated by the points on E , i.e. it can be written as a finite formal sum: $D = \sum_i a_i(P_i)$, where the P_i are points on E and the a_i are integers. In the sequel, we will only consider divisors of degree 0, i.e. such that $\sum_i a_i = 0$.

Given any function f in $K(E)$, we can build a degree 0 divisor $\text{div}(f)$ from the zeros and poles of f simply by forming the formal sum of the zeroes (with multiplicity) minus the formal sum of the poles (with multiplicity). Any divisor $D = \text{div}(f)$ will be called a principal divisor. In the reverse direction, testing whether a degree 0 divisor $D = \sum_i a_i(P_i)$ is principal or not, can be done by evaluating $\sum a_i P_i$ on E . The result will be the point at infinity if and only if D is principal.

Given a function f in $K(E)$ and a point P of E , f can be evaluated at P by substituting the coordinates of P for x and y in any rational map representing f . The function f can also be evaluated at a divisor $D = \sum_i a_i(P_i)$, using the following definition:

$$f(D) = \prod_i f(P_i)^{a_i}.$$

Using these notions, we can now define the Weil pairing: it is a bilinear function from the torsion group $E[n]$ to the multiplicative group μ_n of n -th roots of unity in some extension of \mathbb{F}_p , say \mathbb{F}_{p^k} . Given two n -torsion points P and Q , we compute their pairing $e_n(P, Q)$ by finding two functions f_P and f_Q such that $\text{div}(f_P) = n(P) - n(O)$ and $\text{div}(f_Q) = n(Q) - n(O)$, and by evaluating:

$$e_n(P, Q) = f_P(Q)/f_Q(P).$$

This pairing $e_n : E[n] \times E[n] \rightarrow \mu_n$ is bilinear and non-degenerate. This means that $e_n(aP, bQ) = e_n(P, Q)^{ab}$ and that for some values of P and Q , we have $e_n(P, Q) \neq 1$. We can easily see that given a point X “independent” from P and Q , we can reduce the discrete logarithm problem $Q = \lambda P$ on the elliptic curve to the discrete logarithm problem $e_n(Q, X) = e_n(P, X)^\lambda$ in \mathbb{F}_{p^k} .

The variant of the Tate pairing described in [3] is more complicated, since it operates on divisors instead of points. The Tate pairing operates on n -fold divisors, i.e. divisors D such that nD is principal, it takes values in μ_n and it is bilinear and non-degenerate. Given two n -fold divisors D_1 and D_2 defined over an extension \mathbb{F}_{p^k} that contains the n -th roots of unity, we find f_{D_1} and f_{D_2} such that $\text{div}(f_{D_1}) = nD_1$ and $\text{div}(f_{D_2}) = nD_2$. The Tate pairing of D_1 and D_2 is then defined as:

$$t_n(D_1, D_2) = f_{D_1}(D_2)^{(p^k-1)/n}.$$

This pairing is also bilinear and non-degenerate. Moreover, for the purpose of discrete logarithm reduction, the Tate pairing $t_n(D_1, D_2)$ can easily be transformed into a pairing that involves points. One can simply fix two points R and S , and remark that $t_n((\lambda P) - (O), (R) - (S)) = t_n((P) - (O), (R) - (S))^\lambda$.

For more details about the properties and definitions of the Weil and Tate pairing, we refer the reader to [8,3,4].

3 A Tripartite Diffie–Hellman Protocol

In this section, we want to build an analog of the Diffie–Hellman protocol, that involves three participants A , B and C , requires a single pass of communications and allows the construction of a common secret $K_{A,B,C}$. By a single pass of communication, we mean that each participant is allowed to talk once and broadcast some data to the other two. The main idea is as in ordinary Diffie–Hellman, we start from some elliptic curve E and some point P . Then A , B and C each chose a random number (a , b or c) and they respectively compute $P_A = aP$, $P_B = bP$ and $P_C = cP$ and broadcast these values. Then they respectively compute $F(a, P_B, P_C)$, $F(b, P_A, P_C)$ and $F(c, P_A, P_B)$, where the function F is chosen in a way that ensures that these numbers will be equal and that this common value $K_{A,B,C}$ will be hard to compute given P_A , P_B and P_C . The problem now is to find such an F .

Using the Weil pairing, it is seems very easy to define such an F using the following formula:

$$F_W(x, P, Q) = e_n(P, Q)^x.$$

With this definition, one can easily check that:

$$F_W(a, P_B, P_C) = F_W(b, P_A, P_C) = F_W(c, P_A, P_B) = F_W(1, P, P)^{abc}.$$

However, this function is not satisfying because $e_n(P, P) = 1$ and thus $K_{A,B,C}$ is a constant. Nevertheless, the basic idea is quite sound and can in fact be implemented if we use two independent points P , and Q and if we have the three participants compute and broadcast (P_A, Q_A) , (P_B, Q_B) and (P_C, Q_C) . Then A , B and C can respectively compute $F_W(a, P_B, Q_C) = F_W(a, Q_B, P_C)$, $F_W(b, P_A, Q_C) = F_W(b, Q_A, P_C)$ and $F_W(c, P_A, Q_B) = F_W(c, Q_A, P_B)$. Moreover, all these values are equal and thanks to the independence of P and Q , they are not constant.

Moreover, using two points P and Q , it is easy to use the Tate pairing instead of the Weil pairing, and to define another function F as:

$$F_T(x, D_1, D_2) = t_n(D_1, D_2)^x.$$

Then A , B and C can respectively compute:

$$\begin{aligned} F_T(a, (P_B) - (Q_B), (P_C + Q_C) - (O)) &= \\ &\quad F_T(a, (P_C) - (Q_C), (P_B + Q_B) - (O)), \\ F_T(b, (P_A) - (Q_A), (P_C + Q_C) - (O)) &= \\ &\quad F_T(b, (P_C) - (Q_C), (P_A + Q_A) - (O)), \\ F_T(c, (P_B) - (Q_B), (P_A + Q_A) - (O)) &= \\ &\quad F_T(c, (P_A) - (Q_A), (P_B + Q_B) - (O)). \end{aligned}$$

Because of the bilinearity of the pairing, all these numbers are equal and because of the non-degeneracy, their common value

$$F_T(1, (P) - (Q), (P + Q) - (O))^{abc}$$

is not independent from the choice of a , b and c .

Since F_T is based on the Tate pairing, it will be faster to evaluate than F_W (see the general remark about the efficiency of the Tate pairing versus that of the Weil pairing in [4]). Finally, our tripartite Diffie–Hellman protocol can be summarized as follows:

Alice	Bob	Charlie
Choose a	Choose b	Choose c
Compute (P_A, Q_A)	Compute (P_B, Q_B)	Compute (P_C, Q_C)
Broadcast P_A, P_B, P_C and Q_A, Q_B, Q_C .	Compute the common key as:	
$F_T(a, (P_B) - (Q_B), (P_C + Q_C) - (O))$ $F_T(b, (P_A) - (Q_A), (P_C + Q_C) - (O))$ $F_T(c, (P_B) - (Q_B), (P_A + Q_A) - (O))$		

Choice of Parameters and Construction of the Elliptic Curve

For the tripartite Diffie–Hellman protocol to be efficient, we need to choose elliptic curves such that the pairing can be efficiently computed. This means that the group μ_n should be in a small extension \mathbb{F}_p , i.e. k should be small. Moreover, we need to choose two points P and Q such that the pairing will be non-degenerate, this point can easily be checked by testing whether $e_n(P, Q)$ or $t_n((P) - (Q), (P + Q) - (O))$ is 1 or not. Note that when $k \neq 1$ at least one of the points P and Q must be defined over the extension \mathbb{F}_{p^k} rather than over \mathbb{F}_p otherwise the pairing will always be degenerate.

Two kind of curves are very promising for this tripartite Diffie–Hellman: supersingular curves (which leads to $k = 2$ according to the MOV reduction), and curves of trace 2 (which leads to $k = 1$ according to the FR reduction). It might seem strange to use elliptic curves which are known to be weaker than random curves, however, since we are also mixing in exponentiation in \mathbb{F}_{p^k} , we need to choose a large enough p for the discrete logarithm in \mathbb{F}_{p^k} to be hard and then nobody knows how to compute discrete logarithms on the elliptic curve. The first kind of curve, i.e. supersingular curves, is well known and very easy to build. However, curves of trace 2 are not so easy to construct, in fact, we only known how to construct such curve when $p - 1$ is a square or a small multiple of a square (see [5] or for some examples [4]). This is a pity because curves of trace 2 with a squarefree $p - 1$ would allow us to work with a single point over \mathbb{F}_p instead of two which would be very nice and efficient.

4 Efficient Implementation of the Pairing

The main step when computing the Weil or the Tate pairing is given a n -fold divisor $D = (X) - (Y)$, to write the principal divisor nD as the divisor of a (bivariate) function f denoted by $\text{div}(f)$. Then we need to evaluate f at some other point Z . There exists a standard method to do that, which is based on the fact that every divisor can be written as $(P) - (O) + \text{div}(f)$ for some point P and some function f , and that adding two divisors of that form is easy. Indeed, if

$$\begin{aligned} D &= (P) - (O) + \text{div}(f), \\ D' &= (P') - (O) + \text{div}(f') \end{aligned}$$

then

$$D + D' = (P + P') - (O) + \text{div}(ff'g),$$

where $g = l/v$ with l the line through P and P' and v the vertical line through $P + P'$.

As explained in [7], when writing nD as $\text{div}(f)$, f cannot be expressed as an expanded polynomial (which would be exponentially large) but should be kept in factored form. However, even in factored form, writing down f is quite costly. As an example, the data in [4] shows that such a computation took about 40000 seconds for a supersingular curve when using a 50-digit prime p . This is

not acceptable since with supersingular curves we want to work with a prime number of at least 100 or 150 digits.

In fact, a much better approach is to avoid the computation of f and to directly compute $f(Z)$. This is easily done by keeping for each intermediate divisor D the values of P and $f(Z)$ and by forgetting f . Computing $ff'g(Z)$ is easily performed by multiplying $f(Z)$, $f'(Z)$ and $g(Z)$. Thus at each step, we only need to evaluate two linear polynomials, to compute one inverse and to multiply a couple of numbers. Using this approach and the ZEN library [2], we see in the following example that the Tate pairing can be computed in a single second on a Pentium II-400 processor for a supersingular curve defined over a prime field of more than 150 digits.

A Small Example

In this section, we give an example of the tripartite Diffie–Hellman using a supersingular curve. We chose a prime p of more than 512 bits:

$$\begin{aligned} p = & 48267777815770043535044410856360047038953960729113574 \\ & 29530850774144832990078179684573230519991072031530329 \\ & 37333023591271636050696817523671646492380723773419011. \end{aligned}$$

We are working on the supersingular curve defined by $y^2 = x^3 + x$. Since we need to work in an extension field with p^2 elements, we define this field from the irreducible polynomial $x^2 + 1$, and we denote the square root of 1 by the letter i .

Remark that p was chosen in such a way that the large (160 bits) prime q divides $p + 1$, where:

$$q = 593917583375891588584754753148372137203682206097.$$

We then choose our two points P and Q as points of order q :

$$\begin{aligned} P = & (4419030020021957060597995505214357695235725551511568 \\ & 68511701918183168420954869076254808843953176168634019 \\ & 27551006066189692708095924815897927498508535823262371, \\ & 26090947680860922395540330613428690525406329616428470 \\ & 73807303133884126088547738030713042022034220476530186 \\ & 5163480203757570223664606235381540801075563801118751) \end{aligned}$$

$$Q = (4174183901517981791573276838146590144608495183505084 \\ 36411447781417311430237331232958577456865429161040089 \\ 806217226455983348248260335272068783343983410685645620, \\ 85984079438328066829535503806402848425113755688042614 \\ 53460943539888201506845050435386547281506353153165721 \\ 0019063972911218641810155964304683033635085838106425i)$$

Using the Tate pairing we can compute

$$F_T(1, (P) - (Q), (P + Q) - (O)) = \\ 321226044133092484635656769053049333393058975135298190055 \\ 149195187870368117448022160010655718390434221411264718401 \\ 205796045961343192326955779028644235767724655i + \\ 188248671808397625173631034231316372667592199772896982055 \\ 00343908071592466069428853821862865775750098468723289223 \\ 254974186814834824668646542592184808038517084$$

Then for $a = 4$, $b = 7$ and $c = 28$ we compute

$$F_T(a, (bP) - (bQ), (cP + cQ) - (O)) = \\ 21704655273258595020185058036714661585432952223857344835 \\ 67773957210551020200586870416066057916675619991969502192 \\ 64185045830782800156145170386696601496318727119i + \\ 18547967545356005000241995328735966990113791703635028416 \\ 23483761786522135284562773843989027568976094155038271048 \\ 94436481787700370161453899874562738321254026146$$

and we check that indeed

$$F_T(a, (bP) - (bQ), (cP + cQ) - (O)) = F_T(1, (P) - (Q), (P + Q) - (O))^{abc}$$

Each evaluation of F_T took 1 second on a Pentium II-400 PC running under linux, which is very efficient compared to the 40000 seconds (on a Pentium-75) in [4].

5 Security Issues

Clearly in order to be secure the tripartite Diffie–Hellman described here requires the discrete logarithm on the chosen elliptic curve to be hard, and the discrete logarithm in the finite field \mathbb{F}_{p^k} to be hard. Since we placed ourselves in the cases where either the MOV or the FR reduction applies, the hardness of the elliptic

curve discrete log implies the hardness of the finite field discrete log and we can remove the second condition. This is a simple restatement of the fact that when the finite field discrete log, then to solve the elliptic curve discrete log we simply transport the problem in the finite field using the pairing and then solve the problem in the finite field. However, it is not known whether the elliptic curve discrete logarithm on a weak curve is as hard as the discrete logarithm in the corresponding finite field (in the sense of the MOV or FR reduction). In fact, this is a very interesting open problem. Moreover, as in the Diffie–Hellman case this is not the whole story, some Diffie–Hellman like problem and Diffie–Hellman like decision problem should be hard in order to get security.

Quite amusingly, we should note that on curves where either the MOV or FR reduction applies, the usual Diffie–Hellman decision problem is mostly easy. Remember that the usual Diffie–Hellman problem is given a quadruple (g, g^a, g^b, g^c) to decide whether $c = ab$. This problem can also be expressed with the following formulation which is slightly different. Given a quadruple (g, g^a, h, h^b) , decide whether $a = b$. Now on an elliptic curve where the MOV reduction applies, we can easily test for a quadruple (P, aP, Q, bQ) whether $a = b$, it suffices to compute $e_n(aP, Q)$ and $e_n(P, bQ)$ and to compare them. This test works as soon as P and Q are independent (i.e. when $e_n(P, Q) \neq 1$). Of course, in the FR case, such a test also exists. More precisely, one can test for the equality of $t_n((aP) - (O), (\lambda Q) - (Q))$ and $t_n((P) - (O), (\lambda bQ) - (bQ))$, where λ is essentially any constant number (some values of λ are excluded, for example $\lambda = 1$ is not allowed). Note than when P and Q are not independent, the test usually doesn't work, thus some cases of the usual Diffie–Hellman decision problem are still hard on these elliptic curves.

With the current knowledge of elliptic curves, we believe that this system is secure in practice as soon as the discrete logarithm in \mathbb{F}_{p^k} is hard. For the supersingular case ($k = 2$), we think that p should be a 512 bits prime. In the trace 2 case ($k = 1$), we recommend to choose a 1024 bits prime. Moreover, the usual precautions should be taken, i.e. some large prime q should divide the order of the elliptic curve, all the points involved in the computation should be of order q , and we should use the pairing e_q or t_q .

6 Conclusion

In this article, we described a generalization of the Diffie–Hellman protocol to three parties using the Weil or Tate pairing on elliptic curves. We also showed that this pairing can be implemented much more efficiently than previously shown in [4]. Therefore, this new protocol seems quite promising as a new building block to construct new and efficient complex cryptographic protocols. On the other hand, we sincerely hope that people will try to attack it, since finding a weakness in this protocol would certainly give some new insight in the difficulty of the discrete logarithm on elliptic curves.

References

1. M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In A. De Santis, editor, *Advances in Cryptology — EUROCRYPT’94*, volume 950 of *Lecture Notes in Comput. Sci.*, pages 275–286. Springer, 1995.
2. F. Chabaud and R. Lercier. The ZEN library. <http://www.dmi.ens.fr/~zen>.
3. G. Frey and H. Rück. A remark concerning m -divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62:865–874, 1994.
4. R. Harasawa, J. Shikata, J. Suzuki, and H. Imai. Comparing the MOV and FR reductions in elliptic curve cryptography. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT’99*, volume 1592 of *Lecture Notes in Comput. Sci.*, pages 190–205. Springer, 1999.
5. G.-J. Lay and H. Zimmer. Constructing elliptic curves with given group order over large finite fields. In L. Adleman, editor, *Algorithmic Number Theory*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 250–263. Springer, 1994.
6. R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. thèse, École polytechnique, June 1997.
7. A. Menezes. *Elliptic curve public key cryptosystems*. Kluwer Academic Publishers, 1994.
8. A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transaction on Information Theory*, 39:1639–1646, 1993.
9. V. Miller. Use of elliptic curves in cryptography. In H. Williams, editor, *Advances in Cryptology — CRYPTO’85*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 417–428. Springer, 1986.
10. I. Semaev. Evaluation of discrete logarithms in a group of p -torsion points of an elliptic curve in characteristic p . *Mathematics of Computation*, 67:353–356, 1998.
11. N. Smart. The discrete logarithm problem on elliptic curves of trace one. preprint, 1997.

On Exponential Sums and Group Generators for Elliptic Curves over Finite Fields

David R. Kohel¹ and Igor E. Shparlinski²

¹ School of Mathematics and Statistics
University of Sydney, NSW 2006, Australia

kohel@maths.usyd.edu.au

² Department of Computing
Macquarie University, NSW 2109, Australia
igor@mpce.mq.edu.au

Abstract. In the paper an upper bound is established for certain exponential sums, analogous to Gaussian sums, defined on the points of an elliptic curve over a prime finite field. The bound is applied to prove the existence of group generators for the set of points on an elliptic curve over \mathbb{F}_q among certain sets of bounded size. We apply this estimate to obtain a deterministic $O(q^{1/2+\varepsilon})$ algorithm for finding generators of the group in echelon form, and in particular to determine its group structure.

1 Introduction and Notations

Let $q = p^k$ be a prime power and let \mathcal{E} be an elliptic curve over a finite field \mathbb{F}_q of q elements given by a *Weierstrass* equation

$$y^2 + (a_1x + a_3)y = x^3 + a_2x^2 + a_4x + a_6. \quad (1)$$

The set $\mathcal{E}(\mathbb{F}_q)$ of points over \mathbb{F}_q , together with the point O at infinity as identity, forms an Abelian group. The cardinality of $\mathcal{E}(\mathbb{F}_q)$ is N , where

$$|N - q - 1| \leq 2q^{1/2}.$$

Moreover, as a group, $\mathcal{E}(\mathbb{F}_q)$ is isomorphic to $\mathbb{Z}/M \times \mathbb{Z}/L$ for unique integers M and L with $L \mid M$ and $N = ML$. The number M is called the *exponent* of $\mathcal{E}(\mathbb{F}_q)$. Points P and Q in $\mathcal{E}(\mathbb{F}_q)$ are said to be *echelonized generators* if the order of P is M , the order of Q is L , and any point in $\mathcal{E}(\mathbb{F}_q)$ can be written in the form $mP + \ell Q$ with $1 \leq m \leq M$ and $1 \leq \ell \leq L$.

Although there exists a deterministic polynomial time algorithm to find the number of \mathbb{F}_q -rational points N due to R. Schoof [11] (see also [4,5,16] for references to further theoretical and practical improvements of this algorithm), finding the group structure, or equivalently the exponent M , appears to be a much harder problem.

Once the group order N and the factorization of $r = \gcd(q - 1, N)$ are known, there exists a probabilistic algorithm to compute the group structure in

expected polynomial time (see [9,10]). We note that the existence of the *Weil pairing* (see [18]) implies that L divides r . Thus, using the factorization of r and the nondegeneracy of the Weil pairing, the algorithm finds the value of L . The best possible bound on r is $q^{1/2} + 1$, but for a random curve the value of r tends to be small, in which case the algorithm is efficient.

We now describe the exponential sums which are the subject of study in this work. Let P and Q be echelonized generators for $\mathcal{E}(\mathbb{F}_q)$. For a real number z or element of $\mathbb{Z}/n\mathbb{Z}$, we define

$$\mathbf{e}_n(z) = \exp(2\pi iz/n).$$

The group $\Omega = \text{Hom}(\mathcal{E}(\mathbb{F}_q), \mathbb{C}^*)$ of characters on $\mathcal{E}(\mathbb{F}_q)$ can be described by the set:

$$\Omega = \{\omega \mid \omega(mP + \ell Q) = \mathbf{e}_M(am) \mathbf{e}_L(b\ell) \text{ for } 0 \leq a < M, 0 \leq b < L\}.$$

Similarly the group $\Psi = \text{Hom}(\mathbb{F}_q, \mathbb{C}^*)$ of additive characters on \mathbb{F}_q can be described by the set:

$$\Psi = \{\psi \mid \psi(z) = \mathbf{e}_p(\text{Tr}(\alpha z)) \text{ for } \alpha \in \mathbb{F}_q\},$$

where $\text{Tr}(x)$ is the trace of $x \in \mathbb{F}_q$ to \mathbb{F}_p (see Chapter 2 of [8]). The identity elements of the groups Ω and Ψ are called *trivial* characters.

Let $\mathbb{F}_q(\mathcal{E})$ be the function field of the curve \mathcal{E} . It is generated by the functions x and y , satisfying the Weierstrass equation (1) of the curve, and such that $P = (x(P), y(P))$ for each $P \in \mathcal{E}(\mathbb{F}_q) - \{O\}$.

For characters $\omega \in \Omega$ and $\psi \in \Psi$, and a function $f \in \mathbb{F}_q(\mathcal{E})$, we define the sum

$$S(\omega, \psi, f) = \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ f(P) \neq \infty}} \omega(P)\psi(f(P)).$$

In this work we estimate the exponential sums $S(\omega, \psi, f)$. In particular we will be interested in the sums for $f = x$ or $f = y$. The bounds obtained generalize and improve previous bounds from [13,14]. We apply this bound to design a deterministic algorithm to compute the group structure of $\mathcal{E}(\mathbb{F}_q)$ and to find echelonized generators in time $O(q^{1/2+\varepsilon})$.

In the next section we recall some classical results on L -functions of curves, and relate these to $S(\omega, \psi, f)$.

Throughout the paper $\log z$ denotes the natural logarithm of z .

2 L-Functions of Curves

Let \mathcal{C} be an irreducible projective curve over \mathbb{F}_q of genus g . The divisor group is the free abelian group of formal sums of prime places \mathfrak{P} of $\mathbb{F}_q(\mathcal{C})$. For a fixed algebraic closure $\overline{\mathbb{F}}_q$ of \mathbb{F}_q we can identify a prime place \mathfrak{P} with a Galois orbit $\{P_1, \dots, P_d\}$ of points in $\mathcal{C}(\overline{\mathbb{F}}_q)$, and define $d = \deg(\mathfrak{P})$ to be its degree.

A character χ of the divisor group of $\mathbb{F}_q(\mathcal{C})$ is a map to \mathbb{C} , with image in a finite set $\{0\} \cup \mathbf{e}_n(\mathbb{Z})$ and which is a homomorphism to \mathbb{C}^* on divisors with support outside of a finite set of prime places. Associated to χ is a cyclic Galois cover $\pi : \mathcal{X} \rightarrow \mathcal{C}$ and a divisor $f(\chi)$ called the *conductor*, such that π is unramified outside of the support of $f(\chi)$.

We define the following character sums

$$\sigma_m(\chi) = \sum_{\deg \mathfrak{P} \leq m} \deg(\mathfrak{P}) \chi(\mathfrak{P}), \quad m = 1, 2, \dots,$$

taken over all prime places \mathfrak{P} of $\mathbb{F}_q(\mathcal{C})$ of degree $\deg \mathfrak{P} \leq m$. We define an *L-function*

$$L(\mathcal{C}, t, \chi) = \exp \left(\sum_{m=1}^{\infty} \sigma_m(\chi) t^m / m \right),$$

where $\exp : t\mathbb{C}[[t]] \longrightarrow \mathbb{C}[[t]]$ is given by

$$\exp(h(t)) = \sum_{n=0}^{\infty} \frac{h(t)^n}{n!}.$$

The following proposition for $L(\mathcal{C}, t, \chi)$ appears as Theorem A of [2] or Theorem 6 of Chapter 7 of [20].

Proposition 1. *$L(\mathcal{C}, t, \chi)$ is a polynomial of degree*

$$D = 2g - 2 + \deg f(\chi)$$

where $f(\chi)$ is the conductor of χ . If χ is a product of two characters χ_1 and χ_2 which are ramified in disjoint sets of divisors then

$$\deg f(\chi) = \deg f(\chi_1) + \deg f(\chi_2).$$

We remark the second statement is applicable in particular if one of characters is totally unramified.

We next recall the statement of the Riemann Hypothesis for function fields.

Proposition 2. *Let $\vartheta_1, \dots, \vartheta_D$ be zeros of $L(\mathcal{C}, t, \chi)$ in \mathbb{C} . Then*

$$\sigma_m(\chi) = -(\vartheta_1^m + \dots + \vartheta_D^m),$$

and each zero satisfies $|\vartheta_i| = q^{1/2}$.

3 Exponential Sums on Elliptic Curves

We recall the following standard lemma on character groups of abelian groups.

Lemma 1. *Let G be an abelian group and let $\widehat{G} = \text{Hom}(G, \mathbb{C}^*)$ be its dual group. Then for any element χ of \widehat{G} , we have*

$$\frac{1}{|G|} \sum_{g \in G} \chi(g) = \begin{cases} 1, & \text{if } \chi = \chi_0, \\ 0, & \text{if } \chi \neq \chi_0, \end{cases}$$

where $\chi_0 \in \widehat{G}$ is the trivial character.

In particular, we apply the bound to the pairs $\{\Psi, \mathbb{F}_q\}$ and $\{\mathcal{E}(\mathbb{F}_q), \Omega\}$. By the canonical isomorphism of G with the dual of \widehat{G} , the lemma is symmetrical in G and \widehat{G} .

As an immediate application of Lemma 1 we observe that if ψ_0 is the trivial character, then

$$S(\omega, \psi_0, f) = \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ f(P) \neq \infty}} \omega(P) = - \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ f(P) = \infty}} \omega(P).$$

Thus we see that the interesting part of the exponential sum comes from the character $\psi \circ f$, which defines an Artin-Schreier extension of $\mathbb{F}_q(\mathcal{E})$, as studied in Bombieri [2, Section VI]. We also remark that the exponential sums $S(\omega_0, \psi, f)$ with the trivial character $\omega_0 \in \Omega$ have been estimated in [2].

Let f be a nonconstant function on \mathcal{E} . We write the divisor of poles of f as

$$(f)_\infty = \sum_{i=1}^t n_i \mathfrak{P}_i,$$

where, in particular,

$$\deg(f) = \sum_{i=1}^t n_i \deg(\mathfrak{P}_i). \quad (2)$$

In particular, $\deg f = 2$ if $f = x$, and $\deg f = 3$ if $f = y$. With this notation we have the following theorem.

Theorem 1. *The character ω determines an unramified character, and $\psi \circ f$ determines a character of conductor $\sum_{i=1}^t m_i \mathfrak{P}_i$, where $m_i \leq n_i + 1$ with equality if and only if $(n_i, q) = 1$. The exponential sum satisfies the bound*

$$|S(\omega, \psi, f)| \leq \sum_{i=1}^t m_i \deg(\mathfrak{P}_i) q^{1/2}.$$

Proof. The character ω determines an unramified character mapping through $\mathcal{E}(\mathbb{F}_q)$. Specifically, a prime divisor \mathfrak{P} with associated Galois orbit $\{P_1, \dots, P_d\}$ contained in $\mathcal{E}(\overline{\mathbb{F}}_q)$ maps to the point $P = \sum_i P_i$ in $\mathcal{E}(\mathbb{F}_q)$, and we define $\omega(\mathfrak{P}) = \omega(P)$. The character thus defines a Galois character on the unramified cover defined by the isogeny $\mathcal{E} \rightarrow \mathcal{E}$ with kernel $\mathcal{E}(\mathbb{F}_q)$. In particular the character is unramified and its conductor is trivial. Applying Proposition 1 we reduce to the consideration of the conductor of the character defined by $\psi \circ f$.

The character $\psi \circ f$ defines a Galois character associated to an Artin-Schreier extension of \mathcal{E} , as studied in Bombieri [2, Section VI]. In particular the conductor is determined in Theorem 5 of that work. The bound then follows from Proposition 2. \square

In particular, from Theorem 1 and the identity (2) we see that the bound

$$|S(\omega, \psi, f)| \leq 2\deg(f)q^{1/2} \quad (3)$$

holds. If the polar divisor of f has support at a single prime divisor, then we have the stronger bound

$$|S(\omega, \psi, f)| \leq (1 + \deg(f))q^{1/2}.$$

For a subgroup \mathcal{H} of $\mathcal{E}(\mathbb{F}_q)$ we define

$$S_{\mathcal{H}}(\omega, \psi, f) = \sum_{\substack{P \in \mathcal{H} \\ f(P) \neq \infty}} \omega(P)\psi(f(P))$$

Corollary 1. *Let f be a nonconstant function in $\mathbb{F}_q(\mathcal{E})$ and ψ be a nontrivial character, then the bound*

$$|S_{\mathcal{H}}(\omega, \psi, f)| \leq 2\deg(f)q^{1/2}$$

holds.

Proof. Let $\Omega_{\mathcal{H}} \subseteq \Omega$ be the set of characters $\chi \in \Omega$ such that $\ker(\chi)$ contains \mathcal{H} . Then $\Omega_{\mathcal{H}}$ is dual to $\mathcal{E}(\mathbb{F}_q)/\mathcal{H}$, so we may apply Lemma 1. Therefore

$$\begin{aligned} S_{\mathcal{H}}(\omega, \psi, f) &= \frac{1}{|\Omega_{\mathcal{H}}|} \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ f(P) \neq \infty}} \sum_{\chi \in \Omega_{\mathcal{H}}} \chi(P)\omega(P)\psi(f(P)) \\ &= \frac{1}{|\Omega_{\mathcal{H}}|} \sum_{\chi \in \Omega_{\mathcal{H}}} S(\chi \cdot \omega, \psi, f). \end{aligned}$$

Applying the inequality (3), we obtain the desired estimate. \square

4 Distributions of Points in Intervals

We also require the following standard lemma, which appears, for instance, as Problem 11.c in Chapter 3 of [19].

Lemma 2. *For any positive integers n , s , and r we have*

$$\sum_{k=1}^{n-1} \left| \sum_{a=s}^{s+r} \mathbf{e}_n(ak) \right| \leq n(1 + \log n).$$

We define an *interval* I in \mathbb{F}_q to be a subset of the form $B + \alpha[s, \dots, s+r]$ for an additive subgroup B of \mathbb{F}_q , an element $\alpha \in \mathbb{F}_q$, and nonnegative integers s and r .

Lemma 3. *For any interval I in \mathbb{F}_q the bound*

$$\sum_{\psi \in \Psi} \left| \sum_{\beta \in I} \psi(\beta) \right| \leq q(1 + \log p)$$

holds.

Proof. For an additive subgroup $B \subseteq \mathbb{F}_q$, we define $\Psi_B = \{\psi \in \Psi \mid B \subseteq \ker(\psi)\}$, and note that Ψ_B is dual to \mathbb{F}_q/B .

Now suppose $I = B + \alpha[r, \dots, r+s]$, where $B \subseteq \mathbb{F}_q$ is additive subgroup and $\alpha \notin B$. Since $\sum_{\beta \in B} \psi(\beta) = 0$ for all ψ not in Ψ_B , we can express the sum as

$$\sum_{\psi \in \Psi} \left| \sum_{\beta \in I} \psi(\beta) \right| = \sum_{\psi \in \Psi} \left| \sum_{\beta \in B} \psi(\beta) \sum_{k=r}^{r+s} \psi(k\alpha) \right| = |B| \sum_{\psi \in \Psi_B} \left| \sum_{k=r}^{r+s} \psi(k\alpha) \right|.$$

We set $C = B + \alpha\mathbb{F}_p$, and note that $\psi(k\alpha) = 1$ for all ψ in Ψ_C . Therefore

$$\sum_{\psi \in \Psi} \left| \sum_{\beta \in I} \psi(\beta) \right| = |B| |\Psi_C| \sum_{\psi \in \Psi_B / \Psi_C} \left| \sum_{k=r}^{r+s} \psi(k\alpha) \right|.$$

Since $C/B \cong \alpha\mathbb{F}_p$ is cyclic of order p and with dual group Ψ_B/Ψ_C , we can apply Lemma 2 together with $|B||\Psi_C| = q/p$ to obtain the stated bound. \square

For a character $\omega \in \Omega$, a function $f \in \mathbb{F}_q(\mathcal{E})$, and a subset $S \subseteq \mathbb{F}_q$ we define the

$$T(S, f, \omega) = \{P \in \mathcal{E}(\mathbb{F}_q) \mid f(P) \in S \text{ and } \omega(P) \neq 1\}.$$

and denote its cardinality by $T(S, f, \omega)$.

Theorem 2. *Let \mathcal{E} be an elliptic curve over a finite field \mathbb{F}_q , and let f be a function with poles only at O . Then for any interval $I \subset \mathbb{F}_q$ and character ω of order m , the bound*

$$\left| T(I, f, \omega) - N \frac{(m-1)}{m} \frac{|I|}{q} \right| \leq 2(1 + \deg(f))(1 + \log p)q^{1/2}$$

holds.

Proof. Set \mathcal{H} to be the kernel of ω . Applying Lemma 1 we obtain the expression

$$\begin{aligned} T(I, f, \omega) &= \frac{1}{q} \sum_{\beta \in I} \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ P \notin \mathcal{H}}} \left(\sum_{\psi \in \Psi} \psi(f(P) - \beta) \right) \\ &= \frac{1}{q} \sum_{\psi \in \Psi} \sum_{\substack{P \in \mathcal{E}(\mathbb{F}_q) \\ P \notin \mathcal{H}}} \psi(f(P)) \sum_{\beta \in I} \psi(\beta)^{-1} \\ &= \frac{1}{q} \sum_{\psi \in \Psi} (S(\omega_0, \psi, f) - S_{\mathcal{H}}(\omega_0, \psi, f)) \sum_{\beta \in I} \psi(\beta)^{-1}, \end{aligned}$$

where $\omega_0 \in \Omega$ is the trivial character. Separating out the term corresponding to the trivial character $\psi_0 \in \Psi$, we obtain the expression:

$$T(I, f, \omega) - N \frac{(m-1)}{m} \frac{|I|}{q} = \frac{1}{q} \sum_{\substack{\psi \in \Psi \\ \psi \neq \psi_0}} (S(\omega_0, \psi, f) - S_{\mathcal{H}}(\omega_0, \psi, f)) \sum_{\beta \in I} \psi(\beta)^{-1}.$$

Applying Theorem 1 and Lemma 3 we obtain the desired result. \square

Corollary 2. *Let \mathcal{E} be an elliptic curve over a finite field \mathbb{F}_q of characteristic p , and take either $f = x$ if $p \neq 2$ or $f = y$ if $p \neq 3$ in $\mathbb{F}_q(\mathcal{E})$. Then for any interval $I \subset \mathbb{F}_q$ of cardinality greater than $5(1 + \deg(f))(1 + \log p)q^{1/2}$, the set*

$$\mathcal{T}(I, f) = \{P \in \mathcal{E}(\mathbb{F}_q) \mid f(P) \in I\}$$

generates $\mathcal{E}(\mathbb{F}_q)$.

Proof. Since $\deg(x) = 2$ and $\deg(y) = 3$, we observe that the lower bound on I implies that $|I| > |\mathbb{F}_q|$ for $q < 100$. But for all $q > 100$, we note that the bound

$$\frac{q}{N} \leq \frac{q}{q - 2q^{1/2} + 1} < 1.25$$

holds. Applying the bound of the previous theorem, we find that the subset $\mathcal{T}(I, f, \omega)$ of $\mathcal{T}(I, f)$, is nonempty for any nontrivial character ω . Therefore $\mathcal{T}(I, f)$ is contained in no proper subgroup of $\mathcal{E}(\mathbb{F}_q)$. \square

5 The Algorithm

Theorem 3. *Given any $\varepsilon > 0$, there exists an algorithm which, given an elliptic curve \mathcal{E} over \mathbb{F}_q , constructs echelonized generators for $\mathcal{E}(\mathbb{F}_q)$ in time $O(q^{1/2+\varepsilon})$.*

Proof. For q large, the algorithm works by the following steps, and for small q we may solve the problem by any method we choose.

1. Find the group order N of $\mathcal{E}(\mathbb{F}_q)$, and factor it to find the set of all divisors.
2. Construct the set $\mathcal{T}(I, f)$ of points $P \in \mathcal{E}(\mathbb{F}_q)$ with $f(P) \in I$, for an appropriate choice of function f and interval I , such that $\mathcal{T}(I, f)$ contains generators for $\mathcal{E}(\mathbb{F}_q)$.
3. Reduce the generator set to a pair of echelon generators.

The group order can be computed in polynomial time using the method of Schoof [11], with practical improvements by Atkin and Elkies [5]. The order can be factored by trial division in time $O(q^{1/2+\varepsilon})$, but faster algorithms are also available [1,4], so this phase does not present the limiting complexity.

By Corollary 2, if we set f equal to x for $p \neq 2$ or y if $p = 2$, then the set $\mathcal{T}(I, f)$ contains generators for $\mathcal{E}(\mathbb{F}_q)$ for an interval I of size $O(q^{1/2+\delta})$, where $0 < \delta < \varepsilon$. For each $x_0 \in I$ (or $y_0 \in I$), the points (x_0, y_0) in $\mathcal{E}(\mathbb{F}_q)$, if such exist, can be found by solving a quadratic (or cubic) equation. Knowing a quadratic (or cubic) nonresidue, one can extract roots in polynomial time (see [1,4,16]). The nonresidue can be computed, for instance, by the $O(q^{1/4+\delta})$ -algorithm of [15], which finds a primitive root for \mathbb{F}_q . This one time computation has no impact on the complexity of the algorithm. Therefore the complexity of this stage of the algorithm is

$$O(|\mathcal{T}(I, f)|(\log q)^{O(1)}) = O(q^{1/2+\varepsilon}),$$

which defines the complexity of the algorithm.

Using the factorization of the order N , and a set of generators, we can find the exponent M of the group in polynomial time. If P is a point of order m and Q is a point of order n , where $\gcd(n, m) = 1$, then $P + Q$ has order nm . Thus it suffices to produce echelon generators for each subgroup $\mathbb{Z}/r^\mu\mathbb{Z} \times \mathbb{Z}/r^\lambda\mathbb{Z}$, where r is prime and r^μ and r^λ are the largest powers of r dividing M and $L = N/M$, respectively. Finding an element P of order r^μ involves only polynomial time group operations on elements of the set $\mathcal{T}(I, f)$. Likewise a set of generators for the r^λ -torsion group can be produced in polynomial time, by multiplying points in $\mathcal{T}(I, f)$ by an appropriate factor. Setting $P_1 = r^{\mu-\lambda}P$, we take the Weil pairing of P_1 with each element Q of order r^λ to identify an independent generator (see Menezes [10]). The complexity of this step is again

$$O\left(|\mathcal{T}(I, f)|(\log q)^{O(1)}\right) = O\left(q^{1/2+\varepsilon}\right),$$

so the complexity is as asserted. □

6 Remarks

We note that the methods of this paper can be improved or extended in several ways. From the proof of Corollary 2, it is clear that the constant 5 in the bound can be improved to $4 + o(1)$. A more significantly improvement, however, is achieved using standard techniques (see Chalk [3]) to remove the $\log p$ from the bound. In another direction, combining the method of this paper with a

simple sieve method, it is possible to prove results on the distribution of points whose order equals the group exponent. In particular, for curves with cyclic point group $\mathcal{E}(\mathbb{F}_q)$, one obtains results on the distribution of cyclic generators in intervals. Since none of these results have consequence to the final complexity of the algorithm of this paper, we have left these results to comments.

With minimal modification, the results of this paper carry over to a general result on Jacobians of a hyperelliptic curves over \mathbb{F}_q given by an equation of the form $y^2 + a(x)y = b(x)$, where $a(x)$ and $b(x)$ are polynomials over \mathbb{F}_q . More precisely, it is possible to prove bounds on the size of sets of points *on the curve* which generate the group of rational points *on the Jacobian*. For elliptic curves, the Weil pairing is used to prove the independence of generators for the group of rational points [9,10]. Lacking an effective analogue of the Weil pairing, this approach seems to be the only available deterministic method for producing a provable set of elements generating the group.

For finite fields of bounded characteristic there exist deterministic polynomial time algorithms for constructing a polynomial size set of elements containing a primitive element (see [12,13], and also Chapter 2 of [16]). It remains open whether similar improved bounds hold for the group of rational points on elliptic curves over finite fields of small characteristic.

The bounds of exponential sums of this work also have implications for pseudo-random number generators. The bound of Corollary 1 has been used in [17] to show that the elliptic curve analogue of the Naor–Reingold pseudo-random function is uniformly distributed. Our results can also be used to prove that the elliptic curve analogues of the congruential generator of pseudo-random numbers (see [6,7]) produce uniformly distributed sequences.

Acknowledgment

The authors are grateful to Hendrik Lenstra for valuable advice and his generous sharing of ideas on the subject of this work.

References

1. E. Bach and J. Shallit. *Algorithmic Number Theory*. MIT Press, Cambridge MA, 1996.
2. E. Bombieri. On exponential sums in finite fields. *Amer. J. Math* **88**, 1966, pp. 71–105.
3. J. H. H. Chalk. Polynomial congruences over incomplete residue systems modulo k . *Proc. Kon. Ned. Acad. Wetensch.*, **A92** (1989), 49–62.
4. H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, Berlin, 1997.
5. N. Elkies. Elliptic and modular curves over finite fields and related computational issues. *Computational perspectives on number theory (Chicago, IL, 1995)*, Stud. Adv. Math. **7**, Amer. Math. Soc., Providence, RI, 1998, 21–76.
6. G. Gong, T. A. Bernson and D. A. Stinson. Elliptic curve pseudorandom sequence generators. *Research Report CORR-98-53*, Faculty of Math., Univ. of Waterloo, 1998, 1–21.

7. S. Hallgren. Linear congruential generators over elliptic curves. *Preprint CS-94-143*, Dept. of Comp. Sci., Carnegie Mellon Univ., 1994, 1–10.
8. R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge Univ. Press, Cambridge, 1997.
9. A. J. Menezes, T. Okamoto and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *Trans. IEEE Inform. Theory* **39**, 1993, pp. 1639–1646.
10. A. J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Acad. Publ., Boston, MA, 1993.
11. R. J. Schoof. Elliptic curves over finite fields and the computation of square roots Mod p . *Math. Comp.*, **44** (1985), 483–494.
12. V. Shoup. Searching for primitive roots in finite fields. *Math. Comp.*, **58** (1992), 369–380.
13. I. E. Shparlinski. On primitive elements in finite fields and on elliptic curves. *Matem. Sbornik*, **181** (1990), 1196–1206 (in Russian).
14. I. E. Shparlinski. On Gaussian sums for finite fields and elliptic curves. *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **573** (1992), 5–15.
15. I. E. Shparlinski. On finding primitive roots in finite fields. *Theor. Comp. Sci.*, **157** (1996), 273–275.
16. I. E. Shparlinski. *Finite Fields: Theory and Computation*. Kluwer Acad. Publ., North-Holland, 1999.
17. I. E. Shparlinski. On the Naor–Reingold pseudo-random function from elliptic curves. *Appl. Algebra in Engin., Commun. and Computing* (to appear).
18. J. H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, Berlin, 1995.
19. I. M. Vinogradov. *Elements of Number Theory*. Dover Publ., NY, 1954.
20. A. Weil. *Basic of Number Theory*. Springer-Verlag, Berlin, 1974.

Component Groups of Quotients of $J_0(N)$

David R. Kohel¹ and William A. Stein²

¹ University of Sydney

kohel@maths.usyd.edu.au

<http://www.maths.usyd.edu.au:8000/u/kohel/>

² University of California at Berkeley,

was@math.berkeley.edu

<http://shimura.math.berkeley.edu/~was>

Abstract. Let f be a newform of weight 2 on $\Gamma_0(N)$, and let A_f be the corresponding optimal Abelian variety quotient of $J_0(N)$. We describe an algorithm to compute the order of the component group of A_f at primes p that exactly divide N . We give a table of orders of component groups for all f of level $N \leq 127$ and five examples in which the component group is very large, as predicted by the Birch and Swinnerton-Dyer conjecture.

1 Introduction

Let $X_0(N)$ be the Riemann surface obtained by compactifying the quotient of the upper half-plane by the action of $\Gamma_0(N)$. Then $X_0(N)$ has a canonical structure of algebraic curve over \mathbb{Q} ; denote its Jacobian by $J_0(N)$. It is equipped with an action of a commutative ring $\mathbb{T} = \mathbb{Z}[\dots T_n \dots]$ of Hecke operators. For more details on modular curves, Hecke operators, and modular forms see, e.g., [8].

Now suppose that $f = \sum_{n=1}^{\infty} a_n q^n$ is a modular newform of weight 2 for the congruence subgroup $\Gamma_0(N)$. The Hecke operators also act on f by $T_n(f) = a_n f$. The eigenvalues a_n generate an order $R_f = \mathbb{Z}[\dots a_n \dots]$ in a number field K_f . The kernel I_f of the map $\mathbb{T} \rightarrow R_f$ sending T_n to a_n is a prime ideal. Following Shimura [15], we associate to f the quotient $A_f = J_0(N)/I_f J_0(N)$ of $J_0(N)$. Then A_f is an Abelian variety over \mathbb{Q} of dimension $[K_f : \mathbb{Q}]$, with bad reduction exactly at the primes dividing N .

One-dimensional quotients of $J_0(N)$ have been intensely studied in recent years, both computationally and theoretically. The original conjectures of Birch and Swinnerton-Dyer [1,2], for elliptic curves over \mathbb{Q} , were greatly influenced by computations. The scale of these computations was extended and systematized by Cremona in [6].

In another direction, Wiles [20] and Taylor-Wiles [18] proved a special case of the conjecture of Shimura-Taniyama, which asserts that every elliptic curve over \mathbb{Q} is a quotient of some $J_0(N)$; this allowed them to establish Fermat's Last Theorem. The full Shimura-Taniyama conjecture was later proved by Breuil, Conrad, Diamond, and Taylor in [4]. This illustrates the central role played by quotients of $J_0(N)$.

2 Component Groups of A_f

The Néron model \mathcal{A}/\mathbb{Z} of an Abelian variety A/\mathbb{Q} is by definition a smooth commutative group scheme over \mathbb{Z} with generic fiber A such that for any smooth scheme S over \mathbb{Z} , the restriction map

$$\mathrm{Hom}_{\mathbb{Z}}(S, \mathcal{A}) \rightarrow \mathrm{Hom}_{\mathbb{Q}}(S_{\mathbb{Q}}, A)$$

is a bijection. For more details, including a proof of existence, see, e.g., [5].

Suppose that A_f is a quotient of $J_0(N)$ corresponding to a newform f on $\Gamma_0(N)$, and let \mathcal{A}_f be the Néron model of A_f . For any prime divisor p of N , the closed fiber $\mathcal{A}_{f/\mathbb{F}_p}$ is a group scheme over \mathbb{F}_p , which need not be connected. Denote the connected component of the identity by $\mathcal{A}_{f/\mathbb{F}_p}^{\circ}$. There is an exact sequence

$$0 \rightarrow \mathcal{A}_{f/\mathbb{F}_p}^{\circ} \rightarrow \mathcal{A}_{f/\mathbb{F}_p} \rightarrow \Phi_{A_f, p} \rightarrow 0$$

with $\Phi_{A_f, p}$ a finite étale group scheme over \mathbb{F}_p called the *component group* of A_f at p .

The category of finite étale group schemes over \mathbb{F}_p is equivalent to the category of finite groups equipped with an action of $\mathrm{Gal}(\overline{\mathbb{F}}_p/\mathbb{F}_p)$ (see, e.g., [19, §6.4]). The *order* of an étale group scheme G/\mathbb{F}_p is defined to be the order of the group $G(\overline{\mathbb{F}}_p)$. In this paper we describe an algorithm for computing the order of $\Phi_{A_f, p}$, when p exactly divides N .

3 The Algorithm

Let $J = J_0(N)$, fix a newform f of weight-two for $\Gamma_0(N)$, and let A_f be the corresponding quotient of J . Because J is the Jacobian of a curve, it is canonically isomorphic to its dual, so the projection $J \rightarrow A_f$ induces a polarization $A_f^{\vee} \rightarrow A_f$, where A_f^{\vee} denotes the Abelian variety dual of A_f . We define the *modular degree* δ_{A_f} of A_f to be the positive square root of the degree of this polarization. This agrees with the usual notion of modular degree when A_f is an elliptic curve.

A *torus* T over a field k is a group scheme whose base extension to the separable closure k_s of k is a finite product of copies of \mathbb{G}_m . Every commutative algebraic group over k admits a unique maximal subtorus, defined over k , whose formation commutes with base extension (see IX §2.1 of [9]). The *character group* of a torus T is the group $\mathcal{X} = \mathrm{Hom}_{k_s}(T, \mathbb{G}_m)$ which is a free Abelian group of finite rank together with an action of $\mathrm{Gal}(k_s/k)$ (see, e.g., [19, §7.3]).

We apply this construction to our setting as follows. The closed fiber of the Néron model of J at p is a group scheme over \mathbb{F}_p , whose maximal torus we denote by $T_{J,p}$. We define $\mathcal{X}_{J,p}$ to be the character group of $T_{J,p}$. Then $\mathcal{X}_{J,p}$ is a free Abelian group equipped with an action of both $\mathrm{Gal}(\overline{\mathbb{F}}_p/\mathbb{F}_p)$ and the Hecke algebra \mathbb{T} (see, e.g., [14]). Moreover, there exists a bilinear pairing

$$\langle , \rangle : \mathcal{X}_{J,p} \times \mathcal{X}_{J,p} \rightarrow \mathbb{Z}$$

called the *monodromy pairing* such that

$$\Phi_{J,p} \cong \text{coker}(\mathcal{X}_{J,p} \rightarrow \text{Hom}(\mathcal{X}_{J,p}, \mathbb{Z})).$$

Let $\mathcal{X}_{J,p}[I_f]$ be the intersection of all kernels $\ker(t)$ for t in I_f , and let

$$\alpha_f : \mathcal{X}_{J,p} \rightarrow \text{Hom}(\mathcal{X}_{J,p}[I_f], \mathbb{Z})$$

be the map induced by the monodromy pairing. The following theorem of the second author [16], provides the basis for the computation of orders of component groups.

Theorem 1. *With the notation as above, we have the equality*

$$\#\Phi_{A_f, p} = \frac{\#\text{coker}(\alpha_f) \cdot \delta_{A_f}}{\#(\alpha_f(\mathcal{X}_{J,p})/\alpha_f(\mathcal{X}_{J,p}[I_f]))}.$$

3.1 Computing the Modular Degree δ_{A_f}

Using modular symbols (see, e.g., [6]), we first compute the homology group $H_1(X_0(N), \mathbb{Q}; \text{cusps})$. Using lattice reduction, we then compute the \mathbb{Z} -submodule $H_1(X_0(N), \mathbb{Z}; \text{cusps})$ generated by all Manin symbols (c, d) . Then $H_1(X_0(N), \mathbb{Z})$ is the *integer* kernel of the boundary map.

The Hecke ring \mathbb{T} acts on $H_1(X_0(N), \mathbb{Z})$ and also on $\text{Hom}(H_1(X_0(N), \mathbb{Z}), \mathbb{Z})$, the linear dual, where $t \in \mathbb{T}$ acts on $\varphi \in \text{Hom}(H_1(X_0(N), \mathbb{Z}), \mathbb{Z})$ by $(t \cdot \varphi)(x) = \varphi(tx)$. We have a natural restriction map

$$r_f : \text{Hom}(H_1(X_0(N), \mathbb{Z}), \mathbb{Z})[I_f] \rightarrow \text{Hom}(H_1(X_0(N), \mathbb{Z})[I_f], \mathbb{Z}).$$

Proposition 1. *The cokernel of r_f is isomorphic to the kernel of the polarization $A_f^\vee \rightarrow A_f$ induced by the map $J_0(N) \rightarrow A_f$.*

Thus the order of the cokernel of r_f is the square of the modular degree δ_f . We pause to note that the degree of any polarization is a square; see, e.g., [13, Thm. 13.3].

Proof. Let $S = S_2(\Gamma_0(N), \mathbb{C})$ be the complex vector space of weight-two modular forms of level N , and set $H = H_1(X_0(N), \mathbb{Z})$. The integration pairing $S \times H \rightarrow \mathbb{C}$ induces a natural map

$$\Phi_f : H \rightarrow \text{Hom}(S[I_f], \mathbb{C}).$$

Using the classical Abel-Jacobi theorem, we deduce the following commutative diagram, which has exact columns, but whose rows are not exact.

$$\begin{array}{ccccc}
 & 0 & 0 & 0 & \\
 & \downarrow & \downarrow & \downarrow & \\
 H[I_f] & \longrightarrow & H & \longrightarrow & \Phi_f(H) \\
 \downarrow & & \downarrow & & \downarrow \\
 \text{Hom}(S, \mathbb{C})[I_f] & \longrightarrow & \text{Hom}(S, \mathbb{C}) & \longrightarrow & \text{Hom}(S[I_f], \mathbb{C}) \\
 \downarrow & & \downarrow & & \downarrow \\
 A_f^\vee(\mathbb{C}) & \longrightarrow & J_0(N)(\mathbb{C}) & \longrightarrow & A_f(\mathbb{C}) \\
 \downarrow & \searrow & \downarrow & \nearrow & \downarrow \\
 0 & & 0 & & 0
 \end{array}$$

By the snake lemma, the kernel of $A_f^\vee(\mathbb{C}) \rightarrow A_f(\mathbb{C})$ is isomorphic to the cokernel of the map $H[I_f] \rightarrow \Phi_f(H)$. Since

$$\text{Hom}(H / \ker(\Phi_f), \mathbb{Z}) \cong \text{Hom}(H, \mathbb{Z})[I_f],$$

the $\text{Hom}(-, \mathbb{Z})$ dual of the map $H[I_f] \rightarrow \Phi_f(H) = H / \ker(\Phi_f)$ is r_f , which proves the proposition.

3.2 Computing the Character Group $\mathcal{X}_{J,p}$

Let $N = Mp$, where M and p are coprime. If M is small, then the algorithm of Mestre and Oesterlé [12] can be used to compute $\mathcal{X}_{J,p}$. This algorithm constructs the graph of isogenies between $\overline{\mathbb{F}}_p$ -isomorphism classes of pairs consisting of a supersingular elliptic curve and a cyclic M -torsion subgroup. In particular, the method is elementary to apply when $X_0(M)$ has genus 0.

In general, the above category of “enhanced” supersingular elliptic curves can be replaced by one of left (or right) ideals of a quaternion order \mathcal{O} of level M in the quaternion algebra over \mathbb{Q} ramified at p . This gives an equivalent category, in which the computation of homomorphisms is efficient. The character group $\mathcal{X}_{J,p}$ is known by Deligne-Rapoport [7] to be canonically isomorphic to the degree zero subgroup $\mathcal{X}(\mathcal{O})$ of the free Abelian “divisor group” on the isomorphism classes of enhanced supersingular elliptic curves and of quaternion ideals. Moreover, this isomorphism is compatible with the operation of Hecke operators, which are effectively computable in $\mathcal{X}(\mathcal{O})$ in terms of ideal homomorphisms.

The inner product of two classes in this setting is defined to be the number of isomorphisms between any two representatives. The linear extension to $\mathcal{X}(\mathcal{O})$ gives an inner product which agrees, under the isomorphism, with the monodromy pairing on $\mathcal{X}_{J,p}$. This gives, in particular, an isomorphism $\Phi_{J,p} \cong \text{coker}(\mathcal{X}(\mathcal{O}) \rightarrow \text{Hom}(\mathcal{X}(\mathcal{O}), \mathbb{Z}))$, and an effective means of computing $\#\text{coker}(\alpha_f)$ and $\#(\alpha_f(\mathcal{X}_{J,p}) / \alpha_f(\mathcal{X}_{J,p}[I_f]))$.

The arithmetic of quaternions has been implemented in MAGMA [11] by the first author. Additional details and the application to Shimura curves, generalizing $X_0(N)$, can be found in Kohel [10].

3.3 The Galois Action on $\Phi_{A_f,p}$

To determine the Galois action on $\Phi_{A_f,p}$, we need only know the action of the Frobenius automorphism Frob_p . However, Frob_p acts on $\Phi_{A_f,p}$ in the same way as $-W_p$, where W_p is the p th Atkin-Lehner involution, which can be computed using modular symbols. Since f is an eigenform, the involution W_p acts as either $+1$ or -1 on $\Phi_{A_f,p}$. Moreover, the operator W_p is determined by an involution on the set of quaternion ideals, so it can be determined explicitly on the character group.

4 Tables

The main computational results of this work are presented below in two tables. The relevant algorithms have been implemented in MAGMA and will be made part of a future release. They can also be obtained from the second author.

4.1 Component Groups at Low Level

The first table gives the component groups of the quotients A_f of $J_0(N)$ for $N \leq 127$. The column labeled d contains the dimensions of the A_f , and the column labeled $\#\Phi_{A_f,p}$ contains a list of the orders of the component groups of A_f , one for each divisor p of N , ordered by increasing p . An entry of “?” indicates that $p^2 \mid N$, so our algorithm does not apply. A component group order is starred if the $\text{Gal}(\overline{\mathbb{F}}_p/\mathbb{F}_p)$ -action is nontrivial. More data along these lines can be obtained from the second author.

4.2 Examples of Large Component Groups

Let Ω_{A_f} be the real period of A_f , as defined by J. Tate in [17]. The second author computed the rational numbers $L(A_f, 1)/\Omega_{A_f}$ for every newform f of level $N \leq 1500$. The five largest prime divisors occur in the ratios given in the second table. The Birch and Swinnerton-Dyer conjecture predicts that the large prime divisor of the numerator of each special value must divide the order either of some component group $\Phi_{A_f,p}$ or of the Shafarevich-Tate group of A_f . In each instance $\Phi_{A_f,2}$ is divisible by the large prime divisor, as predicted.

5 Further Directions

Further considerations are needed to compute the *group* structure of $\Phi_{A_f,p}$. However, since the action of Frobenius is known, computing the group structure of $\Phi_{A_f,p}$ suffices to determine its structure as a group scheme.

Our methods say nothing about the component group at primes whose *square* divides the level. The free Abelian group on classes of nonmaximal orders of index p at a ramified prime gives a well-defined divisor group. Do the resulting Hecke modules determine the component groups for quotients of level $p^2 M$?

Component groups at low level

N	d	$\#\Phi_{A_f,p}$									
11	1	5		3	13	76	1	?,*	96	1	?,*
14	1	6*,3	54	1	3*,?	77	1	2*,1*		1	?,*
15	1	4*,4		1	3,?		1	3*,2	97	3	1*
17	1	4	55	1	2,2*		1	6,3*		4	8
19	1	3		2	14*,2		2	2,2*	98	1	2*,?
20	1	?,*	56	1	?,*	78	1	16*,5*,1		2	14,?
21	1	4,2*		1	?,*	79	1	1*	99	1	?,*
23	2	11	57	1	2*,1*		5	13		1	?,*
24	1	?,*		1	2,2*	80	1	?,*		1	?,*
26	1	3*,3		1	10,1*		1	?,*		1	?,*
	1	7,1*	58	1	2*,1*	81	2	?	100	1	?,*
27	1	?		1	10,1*	82	1	2*,1*	101	1	1*
29	2	7	59	5	29		2	28,1*		7	25
30	1	4*,3,1*	61	1	1*	83	1	1*	102	1	2*,2*,1*
31	2	5		3	5		6	41		1	6*,6,1*
32	1	?	62	1	4,1*	84	1	?,*		1	8,4,1
33	1	6*,2		2	66*,3		1	?,*	103	2	1*
34	1	6,1*	63	1	?,*	85	1	2*,1		6	17
35	1	3*,3		2	?,*		2	2*,1*	104	1	?,*
	2	8,4*	64	1	?		2	6,1*		2	?,*
36	1	?,*	65	1	1*,1*	86	2	21*,3	105	1	1,1,1
37	1	1*		2	3*,3		2	55,1*		2	10*,2*,2
	1	3		2	7,1*	87	2	5,1*	106	1	4*,1*
38	1	9*,3	66	1	2*,3,1*		3	92*,4		1	5*,1
	1	5,1*		1	4,1*,1*	88	1	?,*		1	24,1*
39	1	2*,2		1	10,5,1		2	?,*		1	3,1*
	2	14,2*	67	1	1	89	1	1*	107	2	1*
40	1	?,*		2	1*		1	2		7	53
41	3	10		2	11		5	11	108	1	?,*
42	1	8,2*,1*	68	2	?,*	90	1	2*,?,3	109	1	1
43	1	1*	69	1	2,1*		1	6,?,1*		3	1*
	2	7		2	22*,2		1	4,?,1		4	9
44	1	?,*	70	1	4,2*,1*	91	1	1*,1*	110	1	7*,1*,3
45	1	?,*	71	3	5		1	1,1		1	3,1*,1*
46	1	10*,1		3	7		2	7,1*		1	5,5,1
47	4	23	72	1	?,*		3	4*,8		2	16*,3,1*
48	1	?,*	73	1	2	92	1	?,*	111	3	10*,2
49	1	?		2	1*		1	?,*		4	?
50	1	1*,?		2	3	93	2	4*,1*		4	266,2*
	1	5,?	74	2	9*,3		3	64,2*	112	1	?,*
51	1	3,1*		2	95,1*	94	1	2,1*		1	?,*
	2	16*,4	75	1	1*,?		2	94*,1	113	1	2
52	1	?,*		1	1,?	95	3	10,2*		2	2
53	1	1*		1	5,?		4	54*,6		3	1*

Large $L(A_f, 1)/\Omega_{A_f}$

N	dim	$L(A_f, 1)/\Omega_{A_f}$	$\#\Phi_{A_f, p}$
1154 = 2 · 577	20	$2^7 \cdot 85495047371/17^2$	$2^7 \cdot 17^2 \cdot 85495047371, 2^7$
1238 = 2 · 619	19	$2^7 \cdot 7553329019/5 \cdot 31$	$2^7 \cdot 5 \cdot 31 \cdot 7553329019, 2^7$
1322 = 2 · 661	21	$2^7 \cdot 57851840099/331$	$2^7 \cdot 331 \cdot 57851840099, 2^7$
1382 = 2 · 691	20	$2^7 \cdot 37 \cdot 1864449649/173$	$2^7 \cdot 37 \cdot 173 \cdot 1864449649, 2^7$
1478 = 2 · 739	20	$2^7 \cdot 7 \cdot 29 \cdot 1183045463/5 \cdot 37$	$2^7 \cdot 5 \cdot 7 \cdot 29 \cdot 37 \cdot 1183045463, 2^7$

Is it possible to define quantities as in Theorem 1 even when the weight of f is greater than 2? If so, how are the resulting quantities related to the Bloch-Kato Tamagawa numbers (see [3]) of the higher weight motive attached to f ?

References

1. B. J. Birch, and H. P. F. Swinnerton-Dyer, *Notes on elliptic curves, I*, J. Reine Angew. Math. **212** (1963), 7–25.
2. B. J. Birch and H. P. F. Swinnerton-Dyer, *Notes on elliptic curves, II*, J. Reine Angew. Math. **218** (1965), 79–108.
3. S. Bloch and K. Kato, *L -functions and Tamagawa numbers of motives*, The Grothendieck Festschrift, Vol. I, Birkhäuser Boston, Boston, MA, 1990, 333–400.
4. C. Breuil, B. Conrad, F. Diamond, and R. Taylor, *On the modularity of elliptic curves over \mathbb{Q}* , in preparation.
5. S. Bosch, W. Lütkebohmert, and M. Raynaud, *Néron models*, Springer-Verlag, Berlin, 1990.
6. J. E. Cremona, *Algorithms for modular elliptic curves*, second ed., Cambridge University Press, Cambridge, 1997.
7. P. Deligne and M. Rapoport, *Les schémas de modules de courbes elliptiques*, In P. Deligne and W. Kuyk, eds., *Modular functions of one variable, Vol. II*, Lecture Notes in Math., **349**, Springer, Berlin, 1973, 143–316.
8. F. Diamond and J. Im, *Modular forms and modular curves*, In V. K. Murty, ed., *Seminar on Fermat's Last Theorem*, Amer. Math. Soc., Providence, RI, 1995, 39–133.
9. A. Grothendieck, *Séminaire de géométrie algébrique du Bois-Marie 1967–1969 (SGA 7 I)*, Lecture Notes in Mathematics, **288**, Springer-Verlag, Berlin-New York, 1972.
10. D. Kohel, *Hecke module structure of quaternions*, In K. Miyake, ed., *Class Field Theory – Its Centenary and Prospect*, The Advanced Studies in Pure Mathematics Series, Math Soc. Japan, to appear.
11. W. Bosma, J. Cannon, and C. Playoust, *The Magma algebra system I: The user language*, J. Symb. Comp., **24** (1997), no. 3-4, 235–265.
12. J.-F. Mestre, *La méthode des graphes. Exemples et applications*, In *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields*, Nagoya University, Nagoya, 1986, 217–242.
13. J. S. Milne, *Abelian Varieties*, In G. Cornell and J. Silverman, eds., *Arithmetic geometry*, Springer, New York, 1986, 103–150,
14. K. A. Ribet, *On modular representations of $\text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ arising from modular forms*, Invent. Math. **100** (1990), no. 2, 431–476.

15. G. Shimura, *On the factors of the jacobian variety of a modular function field*, J. Math. Soc. Japan **25** (1973), no. 3, 523–544.
16. W. A. Stein, *Explicit approaches to modular abelian varieties*, Ph.D. thesis, University of California, Berkeley, 2000.
17. J. Tate, *On the conjectures of Birch and Swinnerton-Dyer and a geometric analog*, Séminaire Bourbaki, Vol. 9, Soc. Math. France, Paris, 1995, Exp. No. 306, 415–440.
18. R. Taylor and A. Wiles, *Ring-theoretic properties of certain Hecke algebras*, Ann. of Math. **141** (1995), no. 3, 553–572.
19. W. C. Waterhouse, *Introduction to affine group schemes*, Graduate Texts in Mathematics, **66**, Springer-Verlag, New York-Berlin, 1979
20. A. Wiles, *Modular elliptic curves and Fermat's last theorem*, Ann. of Math. **141** (1995), no. 3, 443–551.

Fast Computation of Relative Class Numbers of CM-Fields

Stéphane Louboutin

Université de Caen, Campus 2
Mathématique et Mécanique, BP 5186
14032 Caen cedex, France
loubouti@math.unicaen.fr

Abstract. Let χ be a nontrivial Hecke character on a (strict) ray class group of a totally real number field \mathbf{L} of discriminant $d_{\mathbf{L}}$. Then, $L(0, \chi)$ is an algebraic number of some cyclotomic number field. We develop an efficient technique for computing the exact values at $s = 0$ of such Abelian Hecke L -functions over totally real number fields \mathbf{L} . Let f_{χ} denote the norm of the finite part of the conductor of χ . Then, roughly speaking, we can compute $L(0, \chi)$ in $O((d_{\mathbf{L}} f_{\chi})^{0.5+\epsilon})$ elementary operations. We then explain how the computation of relative class numbers of CM-fields boils down to the computation of exact values at $s = 0$ of such Abelian Hecke L -functions over totally real number fields \mathbf{L} . Finally, we give examples of relative class number computations for CM-fields of large degrees based on computations of $L(0, \chi)$ over totally real number fields of degree 2 and 6. This paper being an abridged version of [Lou4], the reader will find there all the details glossed over here.

1991 Mathematics Subject Classification: Primary 11R29, 11R21, 11Y35.

Keywords and phrases: CM-field, relative class number, Hecke L -function.

1 Notation

Throughout this paper, we let \mathbf{L} be a totally real number field of degree $m \geq 1$ and \mathcal{F} be an integral ideal of \mathbf{L} . We write $\mathbf{L}_{\mathcal{F}}$ for the set of all totally positive elements α of \mathbf{L} such that $\nu_{\mathcal{P}}(\alpha - 1) \geq \nu_{\mathcal{P}}(\mathcal{F})$ for all primes \mathcal{P} of \mathbf{L} which divide \mathcal{F} . The (strict) ray class group mod \mathcal{F} , which we denote by $\mathbf{R}_{\mathcal{F}}(\mathbf{L})$, is defined to be the quotient of the group of fractional ideals of \mathbf{L} generated by the primes not dividing \mathcal{F} , by the subgroup consisting of all principal ideals (α) with $\alpha \in \mathbf{L}_{\mathcal{F}}$. We let χ denote a primitive character or order $n_{\chi} > 1$ on $\mathbf{R}_{\mathcal{F}}(\mathbf{L})$ and set $f_{\chi} = N_{\mathbf{L}/\mathbf{Q}}(\mathcal{F})$, the norm of the finite part of the conductor of χ . We let $\mathbf{M}_{\chi}/\mathbf{L}$ denote the cyclic extension of degree n_{χ} and conductor \mathcal{F} associated with χ and we let w_{χ} denote the number of roots of unity of \mathbf{M}_{χ} . We set $\zeta_{\chi} = \exp(2\pi i/n_{\chi})$, $\mathbf{Q}(\chi) = \mathbf{Q}(\zeta_{\chi})$. We let $\phi_{\chi} = \phi(n_{\chi})$ and $\mathbf{Z}[\chi] = \mathbf{Z}[\zeta_{\chi}]$ denote the degree and the ring of algebraic integers of the cyclotomic field $\mathbf{Q}(\chi)$, respectively. Finally, for any l relatively prime to n_{χ} we let σ_l denote the \mathbf{Q} -automorphism of $\mathbf{Q}(\chi)$ which is defined by $\sigma_l(\zeta_{\chi}) = \zeta_{\chi}^l$.

2 Computation of $L(0, \chi)$

The first aim of this paper is to develop a practical and efficient technique for computing the exact values at $s = 0$ of such Abelian Hecke L -functions, i.e. for computing the exact values of the rational coordinates of this algebraic number $L(0, \chi)$ in a given basis of this cyclotomic field $\mathbf{Q}(\chi)$ (see Theorem 5 and Remark 1). To compute such exact values we fix a \mathbf{Z} -basis \mathcal{B} of the ring of algebraic integers $\mathbf{Z}[\chi]$ of the cyclotomic field $\mathbf{Q}(\chi)$ generated by the values of χ and we compute the coordinates of $L(0, \chi)$ in this basis \mathcal{B} . Since these coordinates are rational numbers whose denominators are bounded beforehand (see Theorem 1), to compute their exact values we only have to compute sufficiently good numerical approximations of them. By expressing these coordinates as linear combinations of finitely many values $L(0, \chi^l)$ for some $l \geq 1$ (see (6) and (7)), we will reduce the computation of approximations of these coordinates to the computation of sufficiently good approximations of values of several $L(0, \chi^l)$.

For each $\mathcal{I} \in \mathbf{R}_{\mathcal{F}}(\mathbf{L})$, the partial zeta function of \mathcal{I} is defined for $\Re(s) > 1$ by $\zeta_{\mathcal{F}}(\mathcal{I}, s) = \sum_{\mathcal{A}} (N(\mathcal{A}))^{-s}$, where the summation is taken over all integral ideals \mathcal{A} of \mathbf{L} , prime to \mathcal{F} , which belong to the class of \mathcal{I} , and where $N(\mathcal{A})$ denotes the norm from \mathbf{L} to \mathbf{Q} of \mathcal{A} . Siegel and Klingen who proved that $\zeta_{\mathcal{F}}(\mathcal{I}, 0)$ is rational. Let now χ be a primitive character of order $n_{\chi} > 1$ on the (strict) ray class group $\mathbf{R}_{\mathcal{F}}(\mathbf{L})$ modulo \mathcal{F} and let

$$L(s, \chi) = \sum_{\mathcal{I}} \chi(\mathcal{I}) \zeta_{\mathcal{F}}(\mathcal{I}, s) \quad (1)$$

(where \mathcal{I} ranges over a set of representatives of the ray class group modulo \mathcal{F}) be the Abelian Hecke L -series associated to χ . Setting

$$a_n(\chi) = \sum_{N(\mathcal{A})=n} \chi(\mathcal{A}) \quad (2)$$

(this sum ranges over all the non zero integral ideals of \mathbf{L} of norm n) we have

$$L(s, \chi) = \sum_{n \geq 1} \frac{a_n(\chi)}{n^s} \quad (\Re(s) > 1) \quad (3)$$

According to (1) and to Siegel-Klingen's Theorem, $L(0, \chi)$ is in $\mathbf{Q}(\chi)$ and for any rational integer l relatively prime to n_{χ} we have

$$\sigma_l(L(0, \chi)) = L(0, \chi^l). \quad (4)$$

Theorem 1. (See [CS] and [Cas]). It holds $w_{\chi} L(0, \chi) \in \mathbf{Z}[\chi]$.

Let $\mathcal{B} = \{\epsilon_1, \dots, \epsilon_{\phi_{\chi}}\}$ be any \mathbf{Z} -basis of $\mathbf{Z}[\chi]$. Let $\mathcal{B}^{\perp} = \{\theta_1, \dots, \theta_{\phi_{\chi}}\}$ be its dual basis relative to the trace form (see [Lan, Prop. 2 page 58]), hence

$$\text{Tr}_{\mathbf{Q}(\chi)/\mathbf{Q}}(\epsilon_k \theta_l) = \delta_{k,l} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l \end{cases}$$

and set

$$M(\mathcal{B}^\perp) = \max_{\substack{1 \leq l \leq n_\chi, \gcd(l, n_\chi)=1 \\ 1 \leq j \leq \phi_\chi}} |\sigma_l(\theta_j)|. \quad (5)$$

Theorem 2. Let \mathcal{F} be a non zero integral ideal of a number field \mathbf{L} of degree $m \geq 1$. Let χ be a primitive character on the (strict) ray class group $\mathbf{R}_\mathcal{F}(\mathbf{L})$ modulo \mathcal{F} and let $f_\chi = N_{\mathbf{L}/\mathbf{Q}}(\mathcal{F})$ denote the norm of the finite part of the conductor of χ . Let $\mathcal{B} = \{\epsilon_1, \dots, \epsilon_{\phi_\chi}\}$ be a \mathbf{Z} -basis of $\mathbf{Z}[\chi]$ and let $\mathcal{B}^\perp = \{\theta_1, \dots, \theta_{\phi_\chi}\}$ be its dual basis relative to the trace form. Define rational integers $b_\chi(k)$ by

$$w_\chi L(0, \chi) = \sum_{k=1}^{\phi_\chi} b_\chi(k) \epsilon_k \in \mathbf{Z}[\chi] \quad (6)$$

(see Theorem 1). We have

$$b_\chi(k) = w_\chi \sum_{\substack{l=1 \\ \gcd(l, n_\chi)=1}}^{n_\chi} \sigma_l(\theta_k) L(0, \chi^l) \quad (7)$$

and these coordinates $b_\chi(k)$ are rational integers of reasonable size:

$$|b_\chi(k)| \leq 2w_\chi M(\mathcal{B}^\perp) \sqrt{d_{\mathbf{L}} f_\chi} \left(\frac{e}{2\pi m} \log(d_{\mathbf{L}} f_\chi) \right)^m. \quad (8)$$

2.1 Numerical Computation of Approximations of $L(0, \chi)$

Theorem 3. For $m \geq 1$, $j \in \{1, 2\}$, $B > 0$ and $\alpha > 1$ we set

$$K_{m,j}(B) = \frac{1}{2\pi i} \int_{\alpha-i\infty}^{\alpha+i\infty} \Gamma^m(s) \frac{B^{2-2s}}{s - (1/j)} ds$$

(which is real and does not depend on $\alpha > 1$). Then

$$0 \leq K_{m,2}(B) \leq K_{m,1}(B) \leq m e^{-B^{2/m}}, \quad (9)$$

Let χ be a primitive Abelian Hecke character over a totally real field \mathbf{L} of degree m . Set $A_\chi = \sqrt{d_{\mathbf{L}} f_\chi / \pi^m}$. Assume that χ is ramified at all the m infinite places of \mathbf{L} , let the $a_n(\chi)$'s be as in (2) and set

$$S_M(\chi) = \frac{A_\chi}{\pi^{m/2}} \sum_{n=1}^M \frac{a_n(\chi)}{n} K_{m,2}(n/A_\chi) + \frac{A_\chi W_\chi}{\pi^{m/2}} \sum_{n=1}^M \frac{\overline{a_n(\chi)}}{n} K_{m,1}(n/A_\chi).$$

For any positive integer $M \geq (m^2/2)^{m/2} A_\chi$ we have:

$$|L(0, \chi) - S_M(\chi)| \leq \frac{2m}{\pi^{m/2}} A_\chi (\log(Me) + m^2/2)^m e^{-(M/A_\chi)^{2/m}}. \quad (10)$$

It now remains to explain how we compute numerically $K_{m,1}(B)$ and $K_{m,2}(B)$ for $B > 0$. We give a precise result for the case $m = 2$:

Theorem 4. *Let $\gamma = 0.577\ 215\ 664\ 901\ 532 \dots$ denote Euler's constant and set $A_1 = 1$, $A_2 = \pi B$. For $B > 0$, we have:*

$$K_{2,j}(B) = A_j + 4 \sum_{n \geq 0} \left(\gamma + \log B - \frac{1}{2n+3-j} - \sum_{k=1}^n \frac{1}{k} \right) \frac{B^{2n+2}}{(2n+3-j)(n!)^2}$$

and for any integer $M \geq 0$ we have $|R_j(M)| \leq 2B^{2M+3}/(M+1)(M!)^2$ where

$$R_j(M) \stackrel{\text{def}}{=} \sum_{n>M} \left(\gamma + \log B - \frac{1}{2n+3-j} - \sum_{k=1}^n \frac{1}{k} \right) \frac{B^{2n+2}}{(2n+3-j)(n!)^2}.$$

Of course, one must finally know how to compute the coefficients $a_n(\chi)$. Since $n \mapsto a_n(\chi)$ is multiplicative, one needs only explain how to compute $a_{p^k}(\chi)$ (and we refer the reader to Proposition 1 for such an example).

2.2 Numerical Computation of the Exact Values of the $b_\chi(k)$

Theorem 5. *Let $\lambda > 1$, $n > 1$ and a \mathbf{Z} -basis \mathcal{B} of the ring of algebraic integers of the cyclotomic field $\mathbf{Q}(\zeta_n)$ be given. Let $M(\mathcal{B}^\perp)$ be as in (5). Let χ range over the primitive characters of order $n_\chi = n$ on (strict) ray class groups and let the $b_\chi(k)$'s be as in (6). Using (10) and (7) we obtain*

$$b_\chi(k) = w_\chi \sum_{\substack{l=1 \\ \gcd(l, n_\chi)=1}}^{n_\chi} \sigma_l(\theta_k) S_M(\chi^l) + O\left(\frac{\log^m A_\chi}{A_\chi^{\lambda-1}}\right). \quad (11)$$

Therefore, for A_χ large enough, the coordinates $b_\chi(k)$ in the basis \mathcal{B} of the algebraic integer $w_\chi L(0, \chi) \in \mathbf{Z}[\chi]$ are rational integers which can be determined in $O(A_\chi^{0.5+\epsilon})$ elementary operations by computing the $\phi(n)$ approximations $S_M(\chi^l)$ for M equal to the least integer greater than or equal to $A_\chi(\lambda \log A_\chi)^{m/2}$ and for l in the range $1 \leq l \leq n$ and $\gcd(l, n) = 1$.

Remark 1. Here we assume that W_χ is known beforehand, for its computation from its definition requires more than f_χ elementary operations. However, for certain classes of characters W_χ is indeed known beforehand (see [FQ], [Fro], [Lou3], (16) and (18) in Section 4). In the case where W_χ it is not known beforehand we explained in [Lou2, Section 5]) how to compute at the same time numerical approximations of W_χ and $L(0, \chi)$ to end up with a practical technique for computing the exact value of $L(0, \chi)$ which conjecturally requires only $A_\chi^{0.5+\epsilon}$ elementary operations.

3 Relative Class Numbers and L -Functions at $s = 0$

Let \mathbf{N} be a CM-field. Then \mathbf{N} is a totally imaginary number field which is a quadratic extension of its maximal totally real subfield \mathbf{N}^+ . Let n denote the degree of \mathbf{N}^+ . Let $h_{\mathbf{N}}^-$, $Q_{\mathbf{N}} \in \{1, 2\}$ and $w_{\mathbf{N}}$ denote the relative class number of \mathbf{N} , the Hasse unit index of \mathbf{N} and the number of roots of unity in \mathbf{N} , respectively. Let \mathbf{L} be any subfield of \mathbf{N}^+ such that the extension \mathbf{N}/\mathbf{L} is Abelian, and let m denote the degree of \mathbf{L} . We thus have the following lattice of subfields:

$$\mathbf{Q} \xrightarrow{m} \mathbf{L} \xrightarrow{n/m} \mathbf{N}^+ \xrightarrow{2} \mathbf{N}$$

We can always choose $\mathbf{L} = \mathbf{N}^+$. However, the smaller is the degree m of \mathbf{L} the more efficient is our technique for computing $h_{\mathbf{N}}^-$. Therefore, we will choose $\mathbf{L} = \mathbf{Q}$ whenever \mathbf{N} is Abelian, whereas we will choose for \mathbf{L} the only real quadratic subfield of \mathbf{N} over which \mathbf{N} is cyclic whenever \mathbf{N} is a dihedral CM-field. We let $X_{\mathbf{N}/\mathbf{L}}$ denote the group of primitive Abelian Hecke characters associated with the Abelian extension \mathbf{N}/\mathbf{L} and set $X_{\mathbf{N}/\mathbf{L}}^- = X_{\mathbf{N}/\mathbf{L}} \setminus X_{\mathbf{N}^+/\mathbf{L}}$. We have:

$$h_{\mathbf{N}}^- = Q_{\mathbf{N}} w_{\mathbf{N}} \prod_{\chi \in X_{\mathbf{N}/\mathbf{L}}^-} 2^{-m} L(0, \chi) \quad (12)$$

Now, let us say that $\chi' \in X_{\mathbf{N}/\mathbf{L}}$ is equivalent to $\chi \in X_{\mathbf{N}/\mathbf{L}}$ if there exists l relatively prime to the order n_{χ} of χ such that $\chi' = \chi^l$. Notice that if χ' is equivalent to χ then χ' and χ both have the same order and conductor. We let $Y_{\mathbf{N}/\mathbf{L}}^-$ denote any set of representatives of the set of equivalence classes of $X_{\mathbf{N}/\mathbf{L}}^-$ modulo this relation. According to (4) and (12) and noticing that for $\chi \in X_{\mathbf{N}/\mathbf{L}}$ we have $\mathbf{L} \subseteq \mathbf{M}_{\chi} \subseteq \mathbf{N}$ (which implies $w_{\chi} \mid w_{\mathbf{N}}$), we have:

Theorem 6. *For any $\chi \in X_{\mathbf{N}/\mathbf{L}}^-$ we have $w_{\mathbf{N}} L(0, \chi) \in \mathbf{Z}[\chi]$ and*

$$h_{\mathbf{N}}^- = Q_{\mathbf{N}} w_{\mathbf{N}} \prod_{\chi \in Y_{\mathbf{N}/\mathbf{L}}^-} N_{\mathbf{Q}(\chi)/\mathbf{Q}} \left(2^{-m} L(0, \chi) \right), \quad (13)$$

and by computing with large rational integers we can easily compute the (usually very large) exact values of the rational numbers $N_{\mathbf{Q}(\chi)/\mathbf{Q}}(2^{-m} L(0, \chi))$ as soon as we have computed the (small) coordinates $b_{\chi}(k) \in \mathbf{Z}$ of $w_{\chi} L(0, \chi)$

Remark 2. Our present method for computing relative class numbers is much more efficient than the method developed in [Lou2, Theorem 7]. There, we had to compute very good approximations of all the $S_M(\chi^l)$ (defined in Theorem 3) prior to taking their product to deduce the value of a relative class number. Here, we only have to compute fair approximations of these $S_M(\chi^l)$ prior to taking linear combinations of them to deduce the exact values of the coordinates of $L(0, \chi)$. Then, we compute the value of the relative class number by computing the norm of the algebraic number $L(0, \chi)$.

4 Examples

4.1 Relative Class Numbers of Some Dihedral CM-Fields

Let $p \geq 3$ be an odd prime. Let \mathbf{N} be a normal CM-field of degree $4p$ whose Galois group is isomorphic to the dihedral group D_{4p} of order $4p$. Hence, $\mathbf{N} = \mathbf{N}^+ \mathbf{M}$ where \mathbf{M} an imaginary biquadratic bicyclic field and \mathbf{N}^+ is a real dihedral field of degree $2p$, cyclic of degree p over the real quadratic subfield \mathbf{L} of \mathbf{M} . There exists a positive rational integer $f \geq 1$ such that the conductor $\mathcal{F}_{\mathbf{N}^+/\mathbf{L}}$ of this extension \mathbf{N}^+/\mathbf{L} is equal to the ideal (f) of \mathbf{L} (see [Mar] and [LPL]). We proved in [LOO] that $Q_{\mathbf{N}} = Q_{\mathbf{M}}$, $w_{\mathbf{N}} = w_{\mathbf{M}}$ and that $h_{\mathbf{M}}^-$ divides $h_{\mathbf{N}}^-$. Hence, formula (13) applied to both \mathbf{N} and \mathbf{M} yields

$$h_{\mathbf{N}}^-/h_{\mathbf{M}}^- = N_{\mathbf{Q}(\zeta_p)/\mathbf{Q}}\left(\frac{1}{4}L(0, \chi)\right).$$

Here, χ is any one of the $p - 1$ primitive characters of order $2p$ associated with the cyclic extension \mathbf{N}/\mathbf{L} . Hence, $w_{\chi} = w_{\mathbf{N}} = w_{\mathbf{M}}$ divides 12. Choose $\mathcal{B} = \{\zeta_p, \dots, \zeta_p^{p-1}\}$. Then $\mathcal{B}^\perp = \{(\zeta_p^{-1} - 1)/p, \dots, (\zeta_p^{-(p-1)} - 1)/p\}$ and

$$w_{\mathbf{M}} L(0, \chi) = \sum_{k=1}^{p-1} b_{\chi}(k) \zeta_p^k \in \mathbf{Z}[\chi] = \mathbf{Z}[\zeta_p] \quad (14)$$

$$\text{with } b_{\chi}(k) = \frac{w_{\mathbf{M}}}{p} \sum_{\substack{l=0 \\ l \neq (p-1)/2}}^{p-1} (\zeta_p^{-k(2l+1)} - 1) L(0, \chi^{2l+1}) \quad (15)$$

(use (7)). Since the induced characters $(\chi^{2l+1})^*$ of the dihedral group $\text{Gal}(\mathbf{N}/\mathbf{Q})$ of order $4p$ are real valued, we have $b_{\chi}(p-k) = b_{\chi}(k)$ for $1 \leq k \leq (p-1)/2$, $L(0, \chi^{2(p-1-l)+1}) = L(0, \chi^{2l+1})$ are real, and thanks to the computation of good enough numerical approximations of the $L(0, \chi^{2l+1})$ for $0 \leq l \leq (p-3)/2$ we can use (15) to compute the exact values of the coordinates $b_{\chi}(k)$ of $L(0, \chi) \in \mathbf{Q}(\zeta_p)^+ = \mathbf{Q}(\cos(2\pi/p))$ and $h_{\mathbf{N}}^-/h_{\mathbf{M}}^- \stackrel{\text{def}}{=} (h_{\mathbf{N}/\mathbf{M}}^-)^2$ is a perfect square with

$$h_{\mathbf{N}/\mathbf{M}}^- = N_{\mathbf{Q}(\zeta_p)^+/\mathbf{Q}}\left(\frac{1}{4}L(0, \chi)\right).$$

Moreover, we have (see [FQ]):

$$W_{\chi} = +1. \quad (16)$$

To make our construction of χ easy, we will choose an example such that

(i) \mathbf{L} has class number one,

(ii) \mathbf{M}/\mathbf{L} is unramified at all the finite places,

(iii) the conductor (f_+) of the cyclic extension \mathbf{N}^+/\mathbf{L} of degree p is of the form (q) for some prime rational number q .

In that situation, χ is a primitive Abelian Hecke character of order $2p$ on the ray class group of conductor $\mathcal{F} = (q)$ of \mathbf{L} and there exists a character χ_+ on

$(\mathbf{A}_L/(q))^*$ of order p (and trivial on the image of \mathbf{Z} in this group) such that for any $\alpha \neq 0$ in the ring of algebraic integers \mathbf{A}_L of L we have $\chi((\alpha)) = \nu(\alpha)\chi_+(\alpha)$, where $\nu(\alpha)$ is the sign of the norm of α .

Example. Choose $p = 41$, $L = \mathbf{Q}(\sqrt{69})$, let N^+ be the only real dihedral field of degree $2p = 82$ for which $\mathcal{F}_{N^+/L} = (q) = (2297)$ and take $M = \mathbf{Q}(\sqrt{-3}, \sqrt{-23})$, for which $h_M^- = 1$. Hence, $N = N^+M$ is a dihedral CM-field of degree $4p = 164$. Since $(q) = \mathcal{QQ}'$ splits in L then (see [Lou2] and [LPL]): $\chi((\alpha)) = \nu(\alpha)\phi(\alpha/\alpha')$ for some character ϕ of order $p = 41$ on the cyclic group $(\mathbf{A}_L/\mathcal{Q})^*$ of order $q-1$, group which is canonically isomorphic to $(\mathbf{Z}/q\mathbf{Z})^*$ (here α' is the conjugate of α in L). To make it explicit which χ we used we chose $\mathcal{Q} = q\mathbf{Z} + ((2527 + \sqrt{d_L})/2)\mathbf{Z}$ and ϕ is the one for which $\phi(5) = \zeta_p = \exp(2\pi i/41)$. According to our numerical computation we have

$$L(0, \chi) = \sum_{k=1}^{40} b_k \zeta_{41}^k$$

with $b_{41-k} = b_k$ and the following Table:

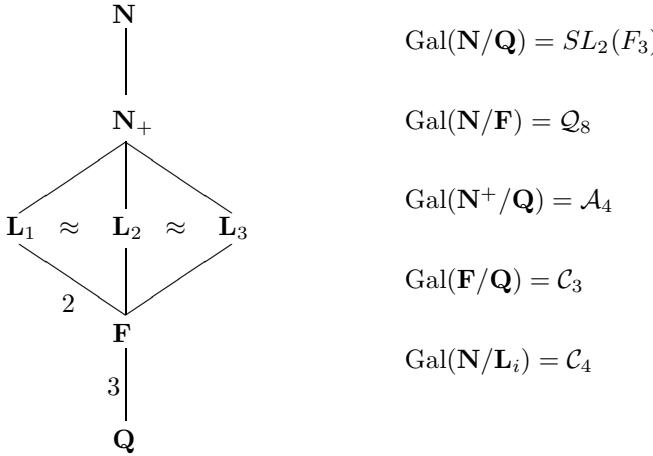
k	1	2	3	4	5
b_k	-4008	-4000	-4028	-4076	-4260
k	6	7	8	9	10
b_k	-4092	-4100	-3964	-3664	-3868
k	11	12	13	14	15
b_k	-3820	-3964	-4024	-4044	-4700
k	16	17	18	19	20
b_k	-4384	-4012	-4068	-3960	-3896

Hence, $h_N^-/h_M^- = (h_{N/M}^-)^2$ with $h_{N/M}^- = 47806\ 51139\ 18289\ 69370\ 25122\ 72645\ 03025\ 58591\ 42700\ 36539\ 28149\ 96559 \approx 4 \cdot 10^{59}$.

4.2 Relative Class Numbers of Some CM-Fields of Degree 24

Let N be a normal CM-field of degree 24 with Galois group isomorphic to $SL_2(F_3)$, the special linear group over the finite field F_3 with three elements (see [Lou1, Section 5] and [LLO]). Then N^+/\mathbf{Q} is a normal extension of degree 12 with Galois group isomorphic to the alternating group A_4 of degree 4 and order 12, and N is a quaternion octic extension of some cyclic cubic field F . Let f_F denote the conductor of F . We let L/F denote a fixed quadratic subextension of the three quadratic subextensions L_i/F of the bicyclic biquadratic extension N^+/F . (Notice that L/\mathbf{Q} is not normal and that the three L_i are conjugate). Then, N/L is a cyclic quartic extension and we let χ denote any one of the two conjugate characters of order four associated with this cyclic quartic extension

N/L. An (incomplete) lattice of subfields is given in the following Diagram:



and we have

$$h_{\mathbf{N}}^- = Q_{\mathbf{N}} w_{\mathbf{N}} N_{\mathbf{Q}(i)/\mathbf{Q}} \left(2^{-6} L(0, \chi) \right).$$

Now, since \mathbf{F} is the maximal Abelian subfield of \mathbf{N} then $w_{\mathbf{N}} = 2$ and $2L(0, \chi) \in \mathbf{Z}[i]$. Since the character of $\text{Gal}(\mathbf{N}/\mathbf{F})$ induced by χ is the irreducible character of degree two of the quaternion octic group which is real valued, then $L(0, \chi)$ is real, hence $L(0, \chi) \in \frac{1}{2}\mathbf{Z}$ and

$$h_{\mathbf{N}}^- = (Q_{\mathbf{N}}/2)(L(0, \chi)/32)^2, \quad (17)$$

which implies $L(0, \chi) \in 32\mathbf{Z}$, and $L(0, \chi) = -A_\chi L(1, \chi)/\pi^3 < 0$. Moreover, if $h_{\mathbf{N}}^-$ is odd then $Q_{\mathbf{N}} = 2$, $h_{\mathbf{N}}^- = (L(0, \chi)/32)^2$ is a perfect square and $L(0, \chi) \notin 64\mathbf{Z}$. Conversely, if $L(0, \chi) \in 32\mathbf{Z} \setminus 64\mathbf{Z}$ then $Q_{\mathbf{N}} = 2$, $h_{\mathbf{N}}^-$ is odd and $h_{\mathbf{N}}^- = (L(0, \chi)/32)^2$ is a perfect square. Now, for any number field \mathbf{E} , let $Cl_{\mathbf{E}}$ and $Cl_{\mathbf{E}}^+$ denote the 2-Sylow subgroups of the ideal class group and narrow ideal class groups of \mathbf{E} , respectively. According to [LLO], if $h_{\mathbf{N}}^-$ is odd then $Cl_{\mathbf{F}}$ and $Cl_{\mathbf{F}}^+$ are both isomorphic to $(\mathbf{Z}/2\mathbf{Z})^2$, $Cl_{\mathbf{L}}$ is isomorphic to $\mathbf{Z}/2\mathbf{Z}$, $Cl_{\mathbf{L}}^+$ is isomorphic to $\mathbf{Z}/4\mathbf{Z}$ and \mathbf{N}^+ is the Hilbert 2-class field of \mathbf{F} and \mathbf{N} is the Hilbert 2-class field of \mathbf{L} . In particular, $A_\chi = f_{\mathbf{F}}^2/\pi^3$ and

$$W_\chi = -1 \quad (18)$$

(for the Abelian extension \mathbf{N}/\mathbf{L} is unramified at all the finite places of \mathbf{L} , but ramified at the six infinite real places of \mathbf{L}).

Now, using class field theory, the reader would prove the following result which, together with the results of section 2.1, enables us to compute the exact values of $L(0, \chi)$ for such characters χ :

Proposition 1. *Let \mathbf{F} , \mathbf{L} , \mathbf{N} and χ be as above. Assume that $f_{\mathbf{F}} = q \equiv 1 \pmod{6}$ is prime, that the narrow and ordinary class groups of \mathbf{F} are isomorphic to $(\mathbf{Z}/2\mathbf{Z})^2$, and that the narrow and ordinary class groups of \mathbf{L} are isomorphic to $\mathbf{Z}/4\mathbf{Z}$ and $\mathbf{Z}/2\mathbf{Z}$, respectively (see [Lou1, Pro. 16]).*

1. if $(p) = \mathbf{P}$ is inert in \mathbf{F} then $(p) = \mathbf{PA_L} = \mathcal{P}\mathcal{P}'$ splits in \mathbf{L}/\mathbf{F} , and setting $\epsilon_p = \chi(\mathcal{P}) = \chi(\mathcal{P}') = \pm 1$, we have

$$a_{p^k}(\chi) = \begin{cases} 0 & \text{if } 3 \text{ does not divide } k \\ \epsilon_p^{k/3}(k+3)/3 & \text{if } 3 \text{ divides } k \end{cases}$$

and $\epsilon_p = +1$ if and only if \mathcal{P} is principal in the narrow sense (notice that \mathcal{P} is always principal in the ordinary sense).

2. if $p = q$ is totally ramified in \mathbf{F} , say $p\mathbf{A_F} = \mathbf{P}^3$, then $\mathbf{PA_L} = \mathcal{P}\mathcal{P}'$ splits in \mathbf{L} , and setting $\epsilon_p = \chi(\mathcal{P}) = \chi(\mathcal{P}') = \pm 1$, we have

$$a_{p^k}(\chi) = \epsilon_p^k(k+1)$$

and $\epsilon_p = +1$ if and only if \mathcal{P} is principal in the narrow sense (notice that \mathcal{P} is always principal in the ordinary sense).

3. Assume that $(p) = \mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ splits in \mathbf{F} . Then either these three ideals are principal in \mathbf{F} or none of them is principal in \mathbf{F} .

- (a) If the three \mathbf{P}_i 's are principal in \mathbf{F} then each $\mathbf{P}_i\mathbf{A_L} = \mathcal{P}_i\mathcal{P}'_i$ splits in \mathbf{L}/\mathbf{F} , and setting $\epsilon_p = \chi(\mathcal{P}_i) = \chi(\mathcal{P}'_i) = \pm 1$ which does not depends on i , we have

$$a_{p^k}(\chi) = \epsilon_p^k(k+5)(k+4)(k+3)(k+2)(k+1)/120$$

and $\epsilon_p = +1$ if and only if \mathcal{P}_i is principal in the narrow sense (notice that \mathcal{P}_i is always principal in the ordinary sense).

- (b) If none of the \mathbf{P}_i is principal in \mathbf{F} then two of these prime ideals $\mathbf{P}_1\mathbf{A_L} = \mathcal{P}_1$ and $\mathbf{P}_2\mathbf{A_L} = \mathcal{P}_2$ are inert in \mathbf{L}/\mathbf{F} and the third one $\mathbf{P}_3\mathbf{A_L} = \mathcal{P}_3\mathcal{P}'_3$ splits in \mathbf{L}/\mathbf{F} . We have $\chi(\mathcal{P}_1) = \chi(\mathcal{P}_2) = -1$, $\chi(\mathcal{P}'_3) = \overline{\chi(\mathcal{P}_3)} = \pm i$ and

$$a_{p^k}(\chi) = \begin{cases} 0 & \text{if } k \text{ is odd} \\ (-1)^{k/2}((k/2)+1)((k/2)+2)/2 & \text{if } k \text{ is even.} \end{cases}$$

For example, using Pari to decide whether a given ideal \mathcal{P} of the sextic field \mathbf{L} is principal in the ordinary or narrow senses, we computed the following Table of relative class numbers:

$f_{\mathbf{F}}$	$P_{\mathbf{F}}(X), P_{\mathbf{L}}(X)$	$L(0, \chi)$	$h_{\mathbf{N}}^-$
163	$P_{\mathbf{F}}(X) = x^3 - x^2 - 54x + 169$	-32	1
	$P_{\mathbf{L}}(X) = x^6 - 3x^5 - 11x^4 + 27x^3 - 3x^2 - 11x + 1$		
349	$P_{\mathbf{F}}(X) = x^3 - x^2 - 116x + 517$	-96	3^2
	$P_{\mathbf{L}}(X) = x^6 - 3x^5 - 17x^4 + 39x^3 - 3x^2 - 17x + 1$		
397	$P_{\mathbf{F}}(X) = x^3 - x^2 - 132x + 544$	-96	3^2
	$P_{\mathbf{L}}(X) = x^6 - 26x^4 + 93x^2 - 4$		
853	$P_{\mathbf{F}}(X) = x^3 - x^2 - 284x - 1011$	-352	11^2
	$P_{\mathbf{L}}(X) = x^6 - 3x^5 - 53x^4 + 111x^3 + 705x^2 - 761x - 91$		
937	$P_{\mathbf{F}}(X) = x^3 - x^2 - 312x + 2221$	-608	19^2
	$P_{\mathbf{L}}(X) = x^6 - 3x^5 - 29x^4 + 63x^3 - 3x^2 - 29x + 1$		

Notice that in these five cases it holds $L(0, \chi) \in 32\mathbf{Z} \setminus 64\mathbf{Z}$. Hence, $Q_{\mathbf{N}} = 2$ (by (17)) and $h_{\mathbf{N}}^- = (L(0, \chi)/32)^2$ is a perfect odd square. Such computations for the 23 CM-fields \mathbf{N} associated with the 23 cyclic cubic fields \mathbf{F} whose conductors $f_{\mathbf{F}}$ are listed in [Lou1, Prop. 16] enable us to prove:

Theorem 7. *There exists only one normal CM-field of degree 24 with Galois group isomorphic to $SL_2(F_3)$ with class number one: the CM-field \mathbf{N} associated with the cyclic cubic field \mathbf{F} of conductor $f_{\mathbf{F}} = 163$.*

4.3 Relative Class Numbers of Some CM-Fields of Degree 42

We refer the reader to [LPCK] for examples of computation of Artin root numbers W_χ and values at $s = 0$ of L -functions associated with characters of order 14 on ray class groups of real cyclic cubic fields \mathbf{L} . These computations are used to prove the following result similar to Theorem 7:

Theorem 8. *There is no non-Abelian normal CM-field of degree 42 with relative class number one.*

References

- Cas. P. Cassou-Noguès. Valeurs aux entiers négatifs des fonctions zêta et fonctions zêta p -adiques. *Invent. Math.* **51** (1979), 29-59.
- CS. J. Coates and W. Sinnott. Integrality properties of the values of partial zeta functions. *Proc. London Math. Soc.* **34** (1977), 365-384.
- FQ. A. Fröhlich and J. Queyrut. On the functional equation of the Artin L -function for characters of real representations. *Invent. Math.* **20** (1973), 125-138.
- Fro. A. Fröhlich. Artin-root numbers and normal integral bases for quaternion fields. *Invent. Math.* **17** (1972), 143-166.
- Hid. H. Hida. *Elementary theory of L -functions and Eisenstein series*. London Mathematical Society, Student Texts **26**, Cambridge University Press, 1993.
- Lan. S. Lang. *Algebraic Number Theory*. Springer-Verlag, Grad. Texts Math. **110**, Second Edition (1994).
- LLO. F. Lemmermeyer, S. Louboutin and R. Okazaki. The class number one problem for some non-Abelian normal CM-fields of degree 24. *Sem. Th. Nb. Bordeaux*, to appear.
- LOO. S. Louboutin, R. Okazaki and M. Olivier. The class number one problem for some non-Abelian normal CM-fields. *Trans. Amer. Math. Soc.* **349** (1997), 3657-3678.
- Lou1. S. Louboutin. Upper bounds on $|L(1, \chi)|$ and applications. *Canad. Math. J.* (4) **50** (1998), 794-815.
- Lou2. S. Louboutin. Computation of relative class numbers of CM-fields by using Hecke L -functions. *Math. Comp.* **69** (2000), 371-393.
- Lou3. S. Louboutin. Formulae for some Artin root numbers. *Tatra Mountains Math. Publ.*, to appear.
- Lou4. S. Louboutin. Computation of $L(0, \chi)$ and of relative class numbers of CM-fields. *Nagoya Math. J.*, to appear.
- LPCK. S. Louboutin, Y.-H. Par, K.-Y. Chang and S.-H. Kwon. The class number one problem for the non Abelian normal CM-fields of degree $2pq$. *Preprint* (1999).
- LPL. S. Louboutin, Y.-H. Park and Y. Lefèuvre. Construction of the real dihedral number fields of degree $2p$. Applications. *Acta Arith.* **89** (1999), 201-215.
- Mar. J. Martinet. Sur l'arithmétique des extensions à groupe de Galois diédral d'ordre $2p$. *Ann. Inst. Fourier (Grenoble)* **19**

On Probable Prime Testing and the Computation of Square Roots mod n

Siguna Müller*

University of Klagenfurt, Dept. of Math., A-9020 Klagenfurt, Austria
`siguna.mueller@uni-klu.ac.at`

Abstract. We will investigate two well-known square root finding algorithms which return the roots of some quadratic residue modulo a prime p . Instead of running the mechanisms modulo p we will investigate their behaviour when applied modulo any integer n . In most cases the results will not be the square roots, when n is composite. Since the results obtained can easily be verified for correctness we obtain a very rapid probable prime test. Based on the square root finding mechanisms we will introduce two pseudoprimality tests which will be shown to be extremely fast and very efficient. Moreover, the proposed test for $n \equiv 1 \pmod{4}$ will be proven to be even more efficient than Grantham's suggestion in [5].

1 Background and Motivation

Two classical problems in number theory at first seem to be unrelated: the computation of square roots modulo a prime p , and the determination of whether or not a given number n is a prime.

A great number of suggestions have been made for efficiently solving these two problems. The operation of *computing square roots modulo p* can be performed with expected running time $O((\lg p)^4)$ (cf. [8]), respectively $O((\lg p)^3)$ (cf. [9]) bit operations. On the other hand, the *primality testing problem* for arbitrary large numbers n is still considered to be a difficult one. Establishing a deterministic answer concerning the primality of any large n seems to be very expensive. Indeed, in practice, probable prime tests are frequently used, which 'only' yield a correct answer with some specific probability that depends on the primality testing condition used.

Most of the pseudoprimality algorithms originate in some sense on **Fermat's Little Theorem** $a^{p-1} \equiv 1 \pmod{p}$, for any base $a \in \mathbb{Z}_p^*$, respectively their specifications based on Euler's criterion, $a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod{p}$, or on the stronger form $a^s \equiv 1$, respectively $a^{2^j s} \equiv -1 \pmod{p}$ for some $0 \leq j \leq r-1$ where $p-1 = 2^r s$ with s odd. Although exponentiation modulo p can be performed extremely fast, the catch of the test is that they allow pseudoprimes, i.e. composite numbers that pass the corresponding conditions.

* Research supported by the Austrian Science Fund (FWF), FWF-Project no. P 13088-MAT

In order to minimize the probability of encountering pseudoprimes, suggestions have been made to replace the testing condition based on the power polynomials with other functions that have suitable properties. These include the use of the roots $\alpha, \bar{\alpha} \in \mathbb{F}_{p^2}$ of some polynomial $f(x) = x^2 - Px + Q$. It turns out, that to some extend this approach can be viewed as an **analogue of the Fermat based tests in terms of arithmetic in the quadratic extension field \mathbb{F}_{p^2}** . Clearly, when p is a prime, $\alpha^{p^2-1} = 1$ in \mathbb{F}_{p^2} where here and in the following $\alpha = \alpha(P, Q)$ denotes any of the two roots in \mathbb{F}_{p^2} . However, it turns out that the exponent $p^2 - 1$ is too large to establish very strong primality criteria.

In particular, a crucial refinement can be made by observing that a condition involving lower exponents can be utilized. In fact, it is well known that $\alpha^{p-(\frac{D}{p})} \equiv 1$, respectively $Q \bmod p$, according as $D = P^2 - 4Q$ is a residue, or nonresidue modulo p . Obviously, the former case reverts us back to the original Fermat condition, so any improvement has to be based on the latter case. The quantity Q , respectively 1, is often called a *general multiplier* (cf. [15]) of α modulo p . Indeed, this additional specific value Q , when $(\frac{D}{p}) = -1$, does seem to play a crucial role in quadratic field based primality testing.

For simplicity we sometimes write $\epsilon(p) = (\frac{D}{p})$, and $\epsilon(n) = (\frac{D}{n})$, when n is any odd integer. Clearly, the condition $\alpha^{n-\epsilon(n)} \equiv Q$, respectively 1 mod n can now be used as a primality testing condition. However, also pseudoprimes based on this approach are known (cf. [3, 5]), also when $\epsilon(n) = -1$. They are often referred to as QF-based (quadratic field based) pseudoprimes, in short *QFpsp(Q)*, or Frobenius pseudoprimes, with respect to the general multiplier Q .

Although the above two major methods of probable prime testing yield a number of pseudoprimes, the combination of these two types of tests is very effective (cf. [2, 5]). No composite number is known yet that passes *both a Fermat based test and a QF-based test* for prescribed parameter searching routines for P, Q (cf. [2]) such that $\epsilon(n) = -1$. Grantham [5] also established strong theoretical results demonstrating that such types of pseudoprimes are very rare.

The combination of the Fermat- and the QF-based tests of course can even be made more powerful when on both sides the stronger versions are being used. This is actually the basis of Grantham's extremely efficient probable prime test. As in the Miller-Rabin test, Grantham's basic idea relies on the fact that the square roots of 1 in the field \mathbb{F}_{p^2} can only be 1 or -1 . Alternatively, we can interpret this as follows. For the roots $\alpha \in \mathbb{F}_{p^2}$ with $\epsilon(p) = -1$ the quantities α^s , respectively $\alpha^{2^j s} \bmod p$ for some $0 \leq j \leq r$ need to be elements of \mathbb{Z}_p^* where now $p+1 = 2^r s$ with s odd. Furthermore, if $(\frac{Q}{p}) = 1$, the latter condition can be sharpened to $j \leq r-1$.

However, it is not known what these values in the prime field \mathbb{F}_p actually need to be. Not even do we know these values in terms of the **generalisation of Euler's criterion in the extension field**. By the Euler condition, $a^{\frac{p-1}{2}}$ has to be equal to the Legendre symbol $(\frac{a}{p})$. On the other hand, for the QF-

based tests, although $\alpha^{p-\epsilon(p)}$ is known to be equal to the general multiplier 1, respectively Q , the immediate question arises, *what the explicit value of $\alpha^{\frac{p-\epsilon(p)}{2}}$ is when reduced modulo a prime p* . In other words, we are interested in the generalisation of the Legendre symbol $\left(\frac{a}{p}\right)$.

This brings us now back to the problem of the computation of square roots. Actually, it is known (cf. [9]) that if Q is a quadratic residue and $\epsilon(p) = -1$ that the quantity $\alpha^{\frac{p-\epsilon(p)}{2}}$ is a square root of Q modulo p . But we don't know which one. Although in the context of finding square roots this does not matter, when applied to pseudoprimality testing, a specific answer would enable us to establish more stringent testing conditions.

Outline of the Paper: In the first part of this contribution we actually exhibit the correct sign and a specific formula for $\alpha^{\frac{p-\epsilon(p)}{2}} \bmod p$ which will indeed turn out to be the exact generalisation of Euler's criterion. That is, for $\epsilon(p) = -1$ we will explicitly evaluate $\alpha^{\frac{p-\epsilon(p)}{2}} \bmod p$ via the 'ordinary' Legendre symbol modulo p with respect to the parameters P, Q .

Consequently, we then utilize properties of square roots modulo primes p as a primality testing condition. We will show that the square root algorithms when applied to any composites n rather than primes will usually not return the correct root modulo n . Furthermore, we introduce two pseudoprimality tests based on the square root finding algorithms that will be shown to be very efficient.

Indeed, the proposed test for $n \equiv 1 \bmod 4$ will be proven to be stronger than Grantham's [5], which presently is the most efficient probable prime test known. Our test has comparable running time, but we will be able to establish a tighter bound on the probability that a composite number fails to be identified as such by the test. Moreover, we believe that our test is easier to describe than Grantham's.

2 Square Roots by Utilizing Quadratic Fields

It is known that the Legendre symbol $\left(\frac{Q}{p}\right)$ can be evaluated very efficiently for any Q coprime to an odd prime p . Consequently, deciding whether or not Q is a square modulo p can be worked out very fast. For actually exhibiting a square root of a quadratic residue Q in \mathbb{F}_p a number of algorithms have been proposed. In detail, if $p \equiv 3 \bmod 4$, this can be achieved very rapidly by means of the square & multiply mechanism. By Euler's criterion we have $Q^{\frac{p-1}{2}} \equiv \left(\frac{Q}{p}\right) \bmod p$, and thus, if $\left(\frac{Q}{p}\right) = 1$, one readily verifies that

$$(\pm Q^{\frac{p+1}{4}}) \bmod p \tag{1}$$

are the two square roots of Q modulo p .

When $p \equiv 1 \bmod 4$, this method obviously cannot be applied. However, when working in the quadratic extension \mathbb{F}_{p^2} of \mathbb{F}_p analogous results may be obtained (cf. [9]).

2.1 Some Fundamental Properties

We will now investigate properties of the elements in the extension field, as this will turn out to yield our fundamental results.

We consider the roots $\alpha, \bar{\alpha} \in \mathbb{F}_{p^2}$ of $f(x) = x^2 - Px + Q$ modulo p . Let $D = P^2 - 4Q$ be the discriminant of $f(x)$ and let $\epsilon(p) = \left(\frac{D}{p}\right)$ denote the Legendre symbol. We will assume throughout that $\epsilon(p) \neq 0$. As noted above, $\alpha^{p-\epsilon(p)} \equiv Q$, respectively $1 \pmod{p}$, according as $\epsilon(p) = -1$ or 1 . Moreover, if $\left(\frac{Q}{p}\right) = 1$ then $\alpha^{\frac{p-\epsilon(p)}{2}} \equiv \bar{\alpha}^{\frac{p-\epsilon(p)}{2}} \pmod{p}$, while if $\left(\frac{Q}{p}\right) = -1$, $\alpha^{\frac{p-\epsilon(p)}{2}} \equiv -\bar{\alpha}^{\frac{p-\epsilon(p)}{2}} \pmod{p}$ (cf. [18]). Thus, in order to establish an analogue of Euler, which holds for any of the roots $\alpha, \bar{\alpha}$, it will only make sense to consider the case that $\left(\frac{Q}{p}\right) = 1$.

Our first step is to establish some link between the roots in \mathbb{F}_{p^2} and the elements P and Q in the prime field.

Lemma 1. *Let $\alpha \in \mathbb{F}_{p^2}$ be any root of $x^2 - Px + Q$ and suppose that a is a square root of Q modulo p . Then $(P + 2a)\alpha \equiv (\alpha + a)^2 \pmod{p}$.*

Proof. By Vieta's rule we have $\alpha^2 + Q \equiv \alpha(\alpha + \bar{\alpha})$ and therefore, $(\alpha + a)^2 \equiv aP + 2aa \pmod{p}$, as claimed. \square

When working in the extension field, we thus have found a square root of $(P + 2a)\alpha$. However, our basic interest is the calculation of square roots of quadratic residues Q in the prime field \mathbb{F}_p . That is, we only utilize the arithmetic of the extension to obtain our desired results modulo p . We still need the following.

Lemma 2. *If $\alpha, \bar{\alpha}$ are the roots of $x^2 - Px + Q$ modulo p and a is a square root of Q modulo p then $\alpha + a$ and $\bar{\alpha} + a$ are the roots of $x^2 - (P + 2a)x + (P + 2a)a \equiv 0 \pmod{p}$, respectively.*

Proof. This follows immediately, since the discriminants $P^2 - 4Q$ and $(P + 2a)^2 - 4(P + 2a)a$ of the two characteristic equations under inspection are the same. \square

2.2 The Generalisation of Euler's Criterion

Theorem 1. *If α is any root of $x^2 - Px + Q$, where $a^2 \equiv Q \pmod{p}$ then*

$$\alpha^{\frac{p-\epsilon(p)}{2}} \equiv \begin{cases} \left(\frac{P+2a}{p}\right) \pmod{p}, & \text{if } \epsilon(p) = 1, \\ \left(\frac{P+2a}{p}\right)a \pmod{p}, & \text{if } \epsilon(p) = -1. \end{cases}$$

Proof. Consider the root $\alpha' \equiv \alpha + a \pmod{p}$ which is by Lemma 2 a root of $x^2 - P'x + Q'$, where $P' \equiv (P + 2a)$, $Q' \equiv (P + 2a)a \pmod{p}$. Then, by the Frobenius automorphism we have $(\alpha + a)^{p-\epsilon(p)} \equiv 1$, respectively $(P + 2a)a \pmod{p}$, according as $\epsilon(p) = \left(\frac{D}{p}\right) = \left(\frac{P'^2 - 4Q'}{p}\right) = 1$ or -1 . We now conclude that

$$(\alpha + a)^{p-\epsilon(p)} \equiv ((P + 2a)a)^{\frac{1-\epsilon(p)}{2}} \pmod{p}.$$

By means of Lemma 1 we also have

$$((\alpha + a)^2)^{\frac{p-\epsilon(p)}{2}} \equiv (P + 2a)^{\frac{p-\epsilon(p)}{2}} \alpha^{\frac{p-\epsilon(p)}{2}} \pmod{p},$$

and therefore, when combining these results,

$$(P + 2a)^{\frac{1-\epsilon(p)}{2}} a^{\frac{1-\epsilon(p)}{2}} \equiv (P + 2a)^{\frac{p-\epsilon(p)}{2}} \alpha^{\frac{p-\epsilon(p)}{2}} \pmod{p},$$

which gives the formula above. The result does not depend on which of the roots $\pm a$ of Q modulo p is being selected. Clearly, the above also holds when a is replaced by $-a \pmod{p}$ in the formula above. \square

As $\left(\frac{(P+2a)a \cdot (P-2a)(-a)}{p}\right) = \left(\frac{-a^2 D}{p}\right) = \left(\frac{-1}{p}\right) \epsilon(p)$, one also has the following.

Corollary 1. Let $\epsilon(p) = -1$. Then $\alpha^{\frac{p-\epsilon(p)}{2}} \equiv \left(\frac{P+2a}{p}\right) \equiv \left(\frac{P-2a}{p}\right) \pmod{p}$, when $p \equiv 3 \pmod{4}$, and $\alpha^{\frac{p-\epsilon(p)}{2}} \equiv \left(\frac{P+2a}{p}\right) \equiv -\left(\frac{P-2a}{p}\right) \pmod{p}$, when $p \equiv 1 \pmod{4}$.

In particular, for any P with $\left(\frac{P^2-4Q}{p}\right) = \epsilon(p) = -1$, Theorem 1 now yields the QF-based method for calculating the square root of Q modulo p .

Corollary 2. Let $x^{\frac{p+1}{2}} \equiv b \pmod{(p, x^2 - Px + Q)}$ where $\left(\frac{P^2-4Q}{p}\right) = \epsilon(p) = -1 \pmod{p}$. Then b is a square root of Q modulo p .

3 Applications to Pseudoprimality Testing

3.1 Pseudoprimes to the Basic Square Root Tests

We now apply the above results as a probable prime testing function. If n is indeed an odd prime, then the properties established above need to be fulfilled. On the other hand, n might fulfill these attributes, although being composite. In other words, there might be some composite numbers that do return the square root of Q modulo this integer n . That is, the above algorithms, when applied modulo n , return the same result as if the square root of Q were calculated modulo each prime factor of n and then combined by means of the Chinese Remainder Theorem.

We will now investigate these pseudoprimes with respect to the above square root algorithms (1) and Corollary 2. Most interestingly, they will correspond to those based on the Fermat- and on the QF-based (or, analogously, Lucas based) probable prime tests, respectively (cf. [14]).

For the different notions and the correspondence of the types of pseudoprimes encountered, we will refer to [14] and [6]. As with all probable prime tests, our goal is to establish some testing conditions that yield strong pseudoprimes. That is, the composite numbers that pass the testing condition should satisfy a number of strong and well specified properties.

Lemma 3. Let $Q \in \mathbb{Z}_n^*$ with $\left(\frac{Q}{n}\right) = 1$ where $n \equiv 3 \pmod{4}$ is a composite integer. Then $a \equiv Q^{\frac{n+1}{4}} \pmod{n}$ fulfills $a^2 \equiv Q \pmod{n}$ iff n is an $Epsp(Q)$, i.e. when $Q^{\frac{n-1}{2}} \equiv 1 \pmod{n}$. Moreover, if a is the correct root of Q modulo n , then n is also a $psp(a)$, i.e. $a^{n-1} \equiv 1 \pmod{n}$.

Proof. If the algorithm does calculate the correct root modulo n , then we have $Q^{\frac{n+1}{2}} \equiv a^2 \pmod{n}$, or alternatively $Q^{\frac{n-1}{2}} \equiv 1 \equiv \left(\frac{Q}{n}\right) \pmod{n}$. The conversion of the statement, as well as the second assertion, is immediately obvious. \square

The underlying fact of our investigations in the quadratic field relies on the property that although when the roots $\alpha, \bar{\alpha}$ of $x^2 - Px + Q$ are elements of the quadratic extension (i.e., when $\epsilon(p) = -1$), certain powers thereof yield elements in the ground field \mathbb{F}_p .

Generally, for $q = p^k$, any integer R such that $\alpha^R = \bar{\alpha}^R = S$ for some nonzero element S in \mathbb{F}_q , is called a **general restricted period** with respect to P and Q . The element S is called a **general multiplier** corresponding to R (cf. [15]).

These two quantities have been specified above modulo primes p . When working modulo composite numbers n we will show below that the corresponding criteria actually provide good primality testing conditions.

Lemma 4. Suppose n is a composite integer, $\left(\frac{Q}{n}\right) = 1$, and $\left(\frac{D}{n}\right) = -1 \pmod{n}$. If $b \equiv x^{\frac{n+1}{2}} \pmod{(n, x^2 - Px + Q)}$ is such that $b^2 \equiv Q \pmod{n}$ and $\left(\frac{P+2b}{n}\right) = 1$, then $R = \frac{n+1}{2}$ is a general restricted period with respect to P and Q modulo n and $b = \left(\frac{P+2a}{n}\right) a$ is the general multiplier corresponding to R . Consequently, n is a $QFpsp(Q)$, i.e., $\alpha^{n+1} \equiv Q \pmod{n}$.

Proof. By means of the arithmetic modulo any prime factor p of n , and modulo $f(x) = x^2 - Px + Q$ it follows that $\alpha^{\frac{n+1}{2}} \equiv \bar{\alpha}^{\frac{n+1}{2}} \equiv b \pmod{p}$ for both roots α and $\bar{\alpha}$ of $f(x)$. Suppose the test passes for the wrong sign of b . That is, $\left(\frac{P-2b}{n}\right) = 1$ but $x^{\frac{n+1}{2}} \pmod{(n, f(x))}$ equals $+b$. But then $\left(\frac{P^2-4Q}{n}\right) = 1$, which is impossible. \square

3.2 An Efficient Probable Prime Test for $n \equiv 3 \pmod{4}$

As noted above, the best results in pseudoprimality testing can be achieved when combining a Fermat- with a QF-based (or, analogously, a Lucas-based [14]) test for $\epsilon(n) = -1$ (cf. [2, 5]).

The arithmetic of the roots modulo p is intimately related to primality tests in terms of the Lucas U - and V - sequences which then yield the corresponding pseudoprimes. In this vein, n is defined a $Lpsp(P, Q)$, $ELpsp(P, Q)$, $sLpsp(P, Q)$, respectively, exactly in the same manner as n is denoted a $psp(a)$, $Epsp(a)$ and $spsp(a)$, respectively, where the conditions of the power polynomials are replaced by the corresponding ones of the Lucas sequences (cf. [14]). For simplicity we use the above abbreviations to denote Lucas pseudoprimes, Euler Lucas pseudoprimes, strong Lucas pseudoprimes with respect to P, Q respectively, as well

as pseudoprimes, Euler pseudoprimes, strong pseudoprimes w.r.t. the base a , respectively (cf. [6, 14]).

The characterisation of the pseudoprimes in the last section which is based on the square root algorithms above, suggest the following probable prime testing algorithm.

Firstly, suppose that $n \equiv 3 \pmod{4}$.

1. Select randomly integers $P \in \mathbb{Z}_n, Q \in \mathbb{Z}_n^*$ with $\left(\frac{Q}{n}\right) = 1, \left(\frac{D}{n}\right) = \epsilon(n) = -1$.
2. Let $a \equiv \pm Q^{\frac{n+1}{4}} \pmod{n}$.
3. [Is a a square root of Q modulo n ?]
If $a^2 \equiv Q \pmod{n}$ go to step 4, else return ‘ n is composite’.
4. If $x^{\frac{n+1}{2}} \not\equiv \left(\frac{P+2a}{n}\right) a \pmod{n}, x^2 - Px + Q$ return ‘ n is composite’, else return ‘ n is a probable prime’.

Theorem 2. Suppose that a composite integer n passes the test of this section. Then n is simultaneously a $spsp(Q)$, a $QFpsp(Q)$, and an $ELpsp(P, Q)$ with $\epsilon(n) = -1$. Moreover, $\left(\frac{Q}{p}\right) = 1$ for all prime divisors p of n .

Proof. The first assertion follows from above since any $Epsp(Q)$ with $\frac{n-1}{2} \equiv 1 \pmod{2}$ already is a strong pseudoprime. But then $\nu_2(n-1) > \nu_2(\text{ord}_n(Q)) = \nu_2(\text{ord}_p(Q))$ for all $p|n$, where $\nu_2(c)$ denotes the highest power of 2 in the prime decomposition of c . Therefore $\left(\frac{Q}{p}\right) = -1$ is impossible since otherwise $\nu_2(p-1) = \nu_2(\text{ord}_p(Q)) < \nu_2(n-1) = 1$ which cannot be. Further, the general restricted period of Lemma 4 satisfies the conditions of an $ELpsp(P, Q)$ since $\left(\frac{Q}{n}\right) = 1$. \square

Remark 1. The approach of selecting the same base Q in the Fermat test as well as in the Lucas test (as second parameter) was first applied in [11] where also a fast method for finding $P, Q \in \{2, -2\}$ with $\left(\frac{D}{n}\right) = -1$ is proposed when $n \not\equiv 1 \pmod{24}$. Based on this parameter selection, an exhaustive search up to 10^{13} has yielded no composite that is simultaneously a $psp(Q)$ and a $Lpsp(P, Q)$.

Remark 2. Note that in our test above, as also in the test described below, we require parameters P and Q such that both $\left(\frac{Q}{n}\right) = 1$ and $\left(\frac{D}{n}\right) = -1$. For practical reasons, it is essential to be able to find these very rapidly. Although no formula is known that on input n , where n is an arbitrary integer, returns P and Q with the above properties, random search is usually very effective. Indeed, in [5] it is shown that the probability of failing to find such a pair is less than $(3/4)^B$, where $B = 50000$ is the limit in the trial division step (which normally is performed before the actual pseudoprimality test).

Although any pseudoprime to the above test has to fulfill the very strong properties above, it will turn out that the square root finding approach will yield an even more stringent test when $n \equiv 1 \pmod{4}$. This is analogue to the fact that any $Epsp(a)$ for $n \equiv 3 \pmod{4}$ is already a $spsp(a)$.

3.3 An Improved Probable Prime Test for $n \equiv 1 \pmod{4}$

1. Select randomly $P \in \mathbb{Z}_n, Q$ in \mathbb{Z}_n^* such that $\left(\frac{Q}{n}\right) = 1, \left(\frac{D}{n}\right) = -1$.
2. Let $b \equiv x^{\frac{n+1}{2}} \pmod{n, x^2 - Px + Q}$.
3. [Is b a square root of Q modulo n ?]
If $b \notin \mathbb{Z}_n^*$ or if $b^2 \not\equiv Q \pmod{n}$, return ‘ n is composite’.
4. [correct sign?]
If $\left(\frac{P+2b}{n}\right) \neq 1$ return ‘ n is composite’, otherwise return ‘ n is a probable prime’.

Remark 3. If both, $\alpha(P, Q)^{\frac{n+1}{2}} \equiv a \pmod{n}$ and $\left(\frac{P+2a}{n}\right) = 1$, trivially the condition $\alpha(P, Q)^{\frac{n+1}{2}} \equiv \left(\frac{P+2a}{n}\right) a \pmod{n}$ is fulfilled. Then, for the root $-a \pmod{n}$ of Q , necessarily $\left(\frac{P-2a}{n}\right) = -1$ since $\left(\frac{D}{n}\right) = -1$. But then also $\alpha(P, Q)^{\frac{n+1}{2}} \equiv a \equiv \left(\frac{P-2a}{n}\right) (-a) \pmod{n}$, which forces n to fulfill the generalised Euler criterion.

Example 1. As an example of a composite integer that satisfies for both roots $\alpha, \bar{\alpha}$ the original Euler-Lucas condition $\alpha(P, Q)^{\frac{n+1}{2}} \equiv \bar{\alpha}(P, Q)^{\frac{n+1}{2}} \pmod{n}$, but that does not pass the generalised Euler condition, we have $n = 341 = 11 \cdot 31$, $P = 2, Q = 4$, because $U_{\frac{n+1}{2}}(P, Q) \equiv 0 \pmod{p}$, but $\left(\frac{P+2b}{n}\right) = -1$, where $b \equiv V_{\frac{n+1}{2}}(P, Q)/2 \equiv 339 \pmod{n}$.

Definition and remark: For any $Q \equiv (\pm a)^2 \pmod{n}$, because a does not appear in the left hand side of $\alpha(P, Q)^{\frac{n+1}{2}} \equiv \pm a \pmod{n}$, the sign on the right hand side can only depend on P . We will throughout denote $\sigma = \sigma(P)$ the sign corresponding to the square root of the Q ’s that occurs on the right hand side. That is, w.l.o.g. we take $\sigma = 1$ for all those P corresponding to the smaller square roots of the Q ’s on the right hand side, and $\sigma = -1$ for those P corresponding to the larger square roots.

Theorem 3. Suppose there is a composite integer n that passes the above test. Then n is simultaneously an Epsp(Q), a QFpsp(Q) and a sLpsp(P, Q) for $\left(\frac{D}{n}\right) = -1$. Moreover, $\left(\frac{Q}{p}\right) = 1$ for all prime divisors p of n .

Proof. Similarly as above, since n is an ELpsp(P, Q) where now $\frac{n-\epsilon(n)}{2} \equiv 1 \pmod{2}$, we conclude that n is already a sLpsp(P, Q). Clearly n fulfills $U_{\frac{n+1}{2}}(P, Q) \equiv 0 \pmod{n}$. In particular, $\frac{n+1}{2}$ is odd. If we suppose that $\left(\frac{Q}{p}\right) = -1$ for some prime p dividing n , then Theorem 1 of [12] asserts that $U_{\frac{n+1}{2}} \not\equiv 0 \pmod{p}$, a contradiction. \square

Recall that our pseudoprimality test was motivated by the square root finding problem of any residue Q . Actually, **Grantham’s test** is based on different ideas, as shortly indicated in the introduction (cf. also [5]). In short, his test consists of the following, where P and Q are chosen as above:

- (1) Perform trial division by primes up to $\min\{B, \sqrt{n}\}$, where $B = 50000$.
In case of divisibility by one of these primes, declare n to be composite and stop.
- (2) If $\sqrt{n} \in \mathbb{Z}$ declare n to be composite and stop.
- (3) Test if $x^{\frac{n+1}{2}} \pmod{n, x^2 - Px + Q} \in \mathbb{Z}_n$.
- (4) Test if $x^{n+1} \equiv Q \pmod{n, x^2 - Px + Q}$.
- (5) Let $n^2 - 1 = 2^r s$ with s odd. Test if $x^s \equiv 1 \pmod{n, x^2 - Px + Q}$, or
 $x^{2^j s} \equiv -1 \pmod{n, x^2 - Px + Q}$ for some $0 \leq j \leq r - 2$.

When some of the steps (3)-(5) return a negative answer then n is composite, otherwise n is declared a probable prime.

Remark 4. — Note that our test of this section fulfills steps (2) - (4) of Grantham's. However, our conditions are at least as strong as his, since we incorporate the actual square root b in the testing function, which additionally limits the chance for a composite number to pass (cf. also Lemma 5 below).

- To obtain a test that is in each of Grantham's steps stronger than his, we only need to verify that n is additionally a strong probable prime to base b . Observe that if n passes, then it is already a probable prime to base b .
- Grantham's step (5) involves the odd parts of $n^2 - 1$. Consequently, when $n \equiv 1 \pmod{4}$, our condition $x^{\frac{n+1}{2}} \equiv b \pmod{n, x^2 - Px + Q}$ implies Grantham's condition, when n is a strong probable prime to base b . Unfortunately, we cannot extend this idea when $n \equiv 3 \pmod{4}$, as then $\frac{n+1}{2}$ is even. This explains why below only our test for $n \equiv 1 \pmod{4}$ will be further analysed.

From the above we immediately get the following.

Theorem 4. *Suppose there is a composite integer n that passes the proposed test of this section and that is also a spsp(b). Then n passes also the probable prime test of [5].*

4 Analysis of the Proposed Test for $n \equiv 1 \pmod{4}$

4.1 A Sharpened Bound on the Error Probability

In the following, we prove an even smaller bound on the probability that a composite number n is wrongfully identified as prime by our proposed test.

As for Grantham's test, we include trial division of small primes up to $\min(B, \sqrt{n})$, where we choose $B = 50000$, as in [5]. Also, we test, if $\sqrt{n} \in \mathbb{Z}$. For convenience, we will call the test of Theorem 4 in combination with these two steps the *multiplier dependent quadratic field based test (MQFT)*.

Proposition 1. *Let n be an odd integer. If p is a prime such that p^2 divides n , then n passes the MQFT with probability less than $2/p$.*

Proof. We follow the proof of [5], but will get a better bound by a factor of $1/2$. Let k be such that $p^k | n$, but $p^{k+1} \nmid n$.

We will show in Lemma 5 below, there are at most $\frac{p-1}{4}$ parameters P with $\left(\frac{P^2-4Q}{p}\right) = -1$ that pass the test modulo p^k for some Q w.r.t. a fixed $\sigma \in \{1, -1\}$. Similarly, it follows from Theorem 1 of [12] that there are at most $\frac{p-1}{4}$ of parameters P with $\left(\frac{P^2-4Q}{p}\right) = 1$ that can pass for some Q . Each parameter P mod p^k corresponds to (n/p^k) parameters mod n , which gives less than $\frac{p-1}{2} \cdot \frac{n}{p^k}$ parameters P mod n for the fixed σ . Analogously we have the same bound for $-\sigma$. So, in total there are at most $(p-1) \cdot \frac{n}{p^k}$ parameters P for which there exists a Q , such that the pair (P, Q) passes. Since $\left(\frac{Q}{p}\right) = 1$ there are at most $\frac{p^k-1}{2}$ parameters Q modulo p^k , and less than $\frac{p^k-1}{2} \cdot \frac{n}{p^k}$ parameters Q mod n .

Hence, altogether n passes for at most $\frac{p^{k+1}-p^k}{2} \cdot \frac{n^2}{p^{2k}} = \frac{1}{2}(1 - \frac{1}{p}) \frac{n^2}{p^{k-1}}$ pairs (P, Q) . The rest now follows exactly in the same manner as in [5]. \square

Proposition 2. *Let n be an odd composite with $p|n$. There are at most $\frac{p-1}{2}$ pairs (P, Q) mod p with $\left(\frac{P^2-4Q}{p}\right) = 1$ that pass the MQFT.*

Proof. This corresponds to Lemma 2.8 of [5]. The result follows exactly in the same way as in [5] since our test is at least as strong as Grantham's. \square

Consequently, we also obtain Lemma 2.9 of [5] concerning the number of pairs for which $\left(\frac{D}{p}\right) = 1$ for at least one prime factor of n . Similarly, we adopt Corollary 2.10 and Lemma 2.11 of [5] in its direct form.

Lemma 5. *Let $n \equiv 1 \pmod{4}$ be any composite integer that passes the test of Theorem 4 for some elements P and some fixed $Q = Q_0$ with $(\pm a)^2 \equiv Q_0 \pmod{n}$. Denote $\alpha = \alpha(P)$ any root of $x^2 - Px + Q_0$ modulo n and let p^k be any prime, respectively prime power, dividing n . Moreover, let $\sigma \in \{-1, 1\}$ be fixed. Then there are at most $\frac{p-1}{4}$ elements P with $\left(\frac{P^2-4Q_0}{p}\right) = -1$ for which $\alpha(P, Q)^{\frac{n+1}{2}} \equiv \sigma a \pmod{p^k}$.*

Proof. Suppose n passes the test for some P and the fixed Q_0 . Then $U_{\frac{n+1}{2}} \equiv 0 \pmod{n}$ for all these P . As $\frac{n+1}{2}$ is odd, Theorem 1 of [12] implies that there are at most $(\frac{n+1}{2}, p - \left(\frac{P^2-4Q_0}{p}\right)) - 1 \leq \frac{p+1}{2} - 1 = \frac{p-1}{2}$ zeros P of $U_{\frac{n+1}{2}} \equiv 0 \pmod{p^k}$ since $\left(\frac{Q_0}{p}\right) = 1$.

We have to calculate the number of zeros that additionally fulfill $\alpha^{\frac{n+1}{2}} \equiv (\frac{P+2a}{n}) a \equiv (\frac{P-2a}{n})(-a) \pmod{n}$ since $\left(\frac{P^2-4Q}{n}\right) = -1$. Denote σa this fixed quantity modulo n .

Clearly, the number of all zeros mod p^k is at most the number of those with the desired general multiplier. We now show that there are as many zeros P that fulfill $\alpha^{\frac{n+1}{2}} \equiv \sigma a \pmod{p^k}$, as zeros that fulfill $\alpha^{\frac{n+1}{2}} \equiv -\sigma a \pmod{p^k}$.

It is known that if $\rho(P) = \rho(p, P, Q_0)$ is the rank of appearance mod p^k (cf. [4, 15]) then there is a unique element s such that $\alpha^{\rho(P)} \equiv \bar{\alpha}^{\rho(P)} \equiv s \pmod{p^k}$,

where $\bar{\alpha}$ denotes the conjugate root of α . We now write $\frac{n+1}{2} = \rho(P)t(P)$ for each of the zeros P . Then $\alpha^{\rho(P)t(P)} \equiv s^{t(P)} \equiv \sigma a \pmod{p^k}$. Since $\frac{n+1}{2}$ is odd and a multiple of $\rho(P)$ for any zero P , both $\rho(P)$ and $t(P)$ need to be odd. It therefore suffices to show that the number of P with odd rank $\rho(P) = r$ and multiplier s is the same as the number of the P' with same odd rank $\rho(P') = r$ and multiplier $-s$.

Recall that $\alpha = \alpha(P)$ is uniquely determined by P . Now take any zero P with rank r and multiplier s . Then $\alpha(-P) \equiv -\bar{\alpha}(P) \pmod{p^k}$, and thus, $\alpha(-P)^r \equiv (-1)^r \bar{\alpha}(P)^r \equiv -s \pmod{p^k}$ because the rank r is odd. Since $\rho(P) = \rho(-P)$, also $t(P) = t(-P) = t$. Consequently, for any P with $\alpha(P)^{\frac{n+1}{2}} \equiv \alpha(P)^{rt} \equiv s^t \equiv \sigma a \pmod{p^k}$, we have that $\alpha(-P)^{\frac{n+1}{2}} \equiv (-s)^t \equiv -\sigma a \pmod{p^k}$, which proves the desired assertion. \square

Remark 5. Observe the role of the σ of the Lemma. As the parameters P are being varied, σ can take both values $\{1, -1\}$. But then, for each Q , there can only be two classes of P 's. Those with $\alpha(P, Q)^{\frac{n+1}{2}} \equiv a$, and those with $\alpha(P, Q)^{\frac{n+1}{2}} \equiv -a \pmod{n}$.

The importance of the lemma lies in the fact that w.r.t. any $Q \equiv a^2 \pmod{n}$, the quantity σ limits the number of P modulo n that can pass the test. In detail, n passes condition 4 of the MQFT only for those $P \pmod{n}$ for which $\alpha^{\frac{n+1}{2}} \equiv \sigma a \pmod{p_i}$ for the same constant value $\sigma \in \{1, -1\}$ for all $p_i | n$.

Lemma 6. *If $n \equiv 1 \pmod{4}$ is squarefree and has k prime factors where k is odd, n passes the MQFT with probability less than $\frac{1}{2^{3k-2}} + \frac{4}{B^2}$.*

Proof. The quantity $\frac{4}{B^2}$ is the same as in Grantham's proof for the number of pairs with $\left(\frac{D}{p}\right) = 1$ for some prime $p | n$. In the following we can assume that $\left(\frac{D}{p}\right) = -1$ for all $p | n$.

We firstly develop an upper bound for the number of Q 's for which there exists at least some P such that (P, Q) passes the test. Recall that necessarily $\left(\frac{Q}{p}\right) = 1$. So, for each p , the number of $Q \pmod{p}$ is at most $\frac{p-1}{2}$.

Suppose that for all the Q with $\left(\frac{Q}{p}\right) = 1$ for all $p | n$ there exist some parameters $P \equiv P(Q) \pmod{n}$ that pass the test. Then,

$$\alpha(P, Q)^{\frac{n+1}{2}} \equiv \left(\frac{P+2a}{n}\right) a \pmod{n}, \quad (2)$$

and, moreover, $\left(\frac{P+2a}{n}\right) a = \sigma a \pmod{n}$ has to be a correct square root of $Q \pmod{n}$. Further, the proof to Lemma 5 shows that w.r.t. $-P \pmod{n}$ the root $-\sigma a \pmod{n}$ will be a correct square root. Since $n \equiv 1 \pmod{4}$, $\left(\frac{-P+2a}{n}\right) \left(\frac{P+2a}{n}\right) \left(\frac{a^2}{n}\right) = \left(\frac{D}{n}\right) = -1$, so that $(-P, Q)$ will pass the first four steps for the general multiplier $-\sigma a \pmod{n}$.

Let now p be any prime dividing n . Then we have $\alpha(P, Q)^{\frac{n+1}{2}} \equiv a \pmod{p}$ and $\alpha(-P, Q)^{\frac{n+1}{2}} \equiv -a \pmod{p}$ (resp. for the conversed signs on the right hand side).

By hypothesis, this holds for every residue $Q \bmod p$, where the $\pm a \bmod p$ denote the square roots of these Q . Thus, by the assumption that the test passes the first four steps for some parameters P for each of the above Q , we obtain $\frac{p-1}{2}$ incongruent elements $a \bmod p$, and also $\frac{p-1}{2}$ incongruent elements $-a \bmod p$ as images on the right hand side in (2), when reduced $\bmod p$. All these incongruent $a, -a \bmod p$ obviously comprise all $\phi(p)$ elements in \mathbb{Z}_p^* .

By assumption that for all the above Q there exist some P that pass the first four steps of the test, we can argue in the same way. For each prime $q|n$ we again obtain $\phi(q)$ incongruent square roots $\pm a \bmod q$ on the right hand side in (2) for q . In total, we obtain $\prod_{p|n} \phi(p) = \phi(n)$ incongruent square roots mod n of all the above Q . Now notice that each of these square roots has to be a basis for the strong probable prime test. However, the number of these ‘liars’ is at most $\phi(n)/4$.

Consequently, not all the Q ’s with $\left(\frac{Q}{p}\right) = 1$ for all $p|n$ pass the MQFT.

We further investigate the set of the Q ’s for which there exist P such that the test does pass. If we denote S_p this set reduced modulo some prime $p|n$, then S_p obviously is a subgroup of the group of the residues $\bmod p$. This follows since all elements are squares, the above square root finding function is multiplicative, and the square roots of ± 1 tested by the strong probable prime test, are trivially fulfilled modulo a prime p .

By the Chinese Remainder Theorem, the above deduction shows that there has to be at least one prime $p_1|n$ for which the set S_{p_1} has cardinality less than $\frac{p_1-1}{2}$. Consequently, for this p_1 , S_{p_1} has at most $\frac{p_1-1}{4} < \frac{p_1}{4}$ elements.

On the other hand, Lemma 5 asserts that for each of the Q ’s there are less than $\frac{p-1}{4} < \frac{p}{4}$ parameters P that fulfill $\alpha^{\frac{n+1}{2}} \equiv \sigma a \bmod p$ for a fixed $\sigma \bmod n$. Thus, by the Chinese Remainder Theorem, there are less than $(p_1/4) \prod_{p \neq p_1} \frac{p}{2} \cdot \prod_{p|n} \frac{p}{4} = \frac{n^2}{2^{3k+1}}$ pairs for a fixed $\sigma \bmod n$. Finally, for both $\sigma = 1$ and -1 , the number of pairs that pass the MQFT is less than $\frac{n^2}{2^{3k}}$.

The desired probability now follows from the number of possible pairs (P, Q) which is at least $\frac{n^2}{4}$ (cf. [5]). □

We thus have our main results.

Theorem 5. *An odd composite number $n \equiv 1 \bmod 4$ passes the MQFT with probability less than $\frac{1}{8190}$.*

Proof. If n is not squarefree, Proposition 1 gives the result. If a squarefree n has an even number of prime factors we can apply Corollary 2.10 of [5], which gives a probability less than $2/B$. Also, if n is squarefree and has exactly 3 prime factors, we use Lemma 2.11 of [5] which bounds the probability by $\frac{4}{B^2} + \frac{3(B^2+1)}{2(B^4-3B^2)}$.

In the remaining cases we apply Lemma 6 which gives largest probability when $k = 5$ in which case it is bounded by $1/2^{13} + 1/25000^2$. □

Theorem 6. *Suppose that an odd composite integer $n \equiv 1 \bmod 4$ is not one of the following:*

- a product of exactly 5 prime factors,
- a superstrong Dickson pseudoprime of type II (cf. [13]), i.e. one for which $p^2 - 1 \mid n - p$ for all prime factors p of n .

Then n passes the MQFT with probability less than $\frac{1}{25000}$.

Proof. As above we distinguish the cases, n not squarefree, n squarefree and divisible by an even number of primes, n squarefree and divisible by three, respectively an odd number, of prime factors. We only need to improve on the bound of Lemma 5 as the remaining bounds are tight enough. We can assume that $(\frac{D}{p}) = -1$ for all $p \mid n$. Any n that has exactly 5 prime factors yields the bound of Theorem 5, but for a larger number of prime divisors, Proposition 1 and Lemma 5 yield the bound of this Theorem. This proves the first assertion.

Now consider the second case. Then, for any $p \mid n$, $\alpha^{n+1} \equiv \alpha^{p+1} \equiv Q \pmod{p}$ and since Q is invertible, $\alpha^{n-p} \equiv 1 \pmod{p}$. There are $(n-p, p^2-1)$ solutions of $x^{n-p} \equiv 1 \pmod{(p, x^2 - Px + Q)}$. Each pair (P, Q) corresponds to two solutions with minimal polynomial $x^2 - Px + Q$ which either both do or both do not satisfy $x^{n-p} \equiv 1$. Thus, there are at most $\frac{(n-p, p^2-1)}{2} \leq \frac{p^2-1}{4}$ such pairs mod p . As in the proof of Lemma 6, Lemma 5 asserts that for any fixed $\sigma \pmod{n}$ we can only count half of the parameters P (for some Q) modulo each p , so that the number becomes at most $\frac{p^2-1}{8}$. Moreover, we also can count only the Q 's that are squares modulo p , where as in the proof to Lemma 6 for at least one prime $p_1 \mid n$ the set of the Q 's has cardinality less than $p_1/4$. Consequently, for $p \neq p_1$ there are less than $\frac{p^2}{16}$, and for p_1 less than $\frac{p_1^2}{32}$ pairs (P, Q) that can pass the test w.r.t. a fixed sign $\sigma \pmod{n}$, corresponding to the signs of the square roots of the Q 's. Then multiplying over all primes and adding for the two σ 's gives the desired probability. \square

Remark 6. It is widely believed that type II superstrong Dickson pseudoprimes actually cannot exist. However, proving or disproving this claim has turned out to be quite delicate. Theorem 6 cannot be applied when such a number with exactly 5 prime factors does exist. This seems to be extremely unlikely.

4.2 Performance and Practical Considerations

Although the **MQFT** w.r.t. the pair (P, Q) is defined in the context of the field \mathbb{F}_{p^2} , it can be rephrased **in terms of Lucas sequences**, as described below.

Practically, the approach of utilizing Lucas sequences can be realised very fast. In particular, we only require the combined evaluation of both the U - and the V -sequence of same degree. But this can be obtained in almost the same time as the evaluation of the U -sequence by itself (cf. e.g. [17]). Moreover, the main advantage (as opposed to the second method below) is that many fast software packages for the rapid evaluation of Lucas sequences are readily available.

1. Perform trial division by primes up to $\min\{B, \sqrt{n}\}$, as above.
2. If $\sqrt{n} \in \mathbb{Z}$ declare n to be composite and stop.
3. If $U_{\frac{n+1}{2}}(P, Q) \not\equiv 0 \pmod{n}$ declare n to be composite and stop.

4. Let $b \equiv \left(V_{\frac{n+1}{2}}(P, Q)\right) / 2 \pmod n$. If $b^2 \not\equiv Q \pmod n$ then declare n to be composite and stop.
5. If $\left(\frac{P+2b}{n}\right) \neq 1$ declare n to be composite and stop.
6. If n is not a $spsp(b)$ then n is composite, otherwise declare n to be a probable prime.

Nonetheless, the test can be performed even faster by utilizing binary addition/Lucas chains [5, 10]. In particular, if $V_m(P, Q)$ is transformed into a sequence with second parameter equal to 1, then the combined evaluation of $V_m(P, Q)$ and $V_{m+1}(P, Q)$ may be computed modulo n using less than $2 \lg n$ multiplications mod n and $\lg n$ additions mod n . Moreover, as for $m = \frac{n+1}{2}$ the term $U_{2m}(P, Q) \pmod n$ has to vanish, it follows that the time for performing the Lucas tests in the MQFT is about twice the time to do a strong probable prime test. We obtain the same upper bound for the running time as Grantham does.

Lemma 7. *The time to perform the MQFT is about three times the time to do a strong probable prime test.*

As in Grantham's case, this demonstrates the high efficiency of the MQFT. If one does three iterations of the strong probable prime test, a composite number will fail to be recognised as such at most 1/64 of the time. By contrast, in about the same time, the MQFT recognises composites with failure at most 8190 (respectively 25000).

Moreover, although $spsp$'s to quite a large number of different bases are known [1, 7] no composite number that passes Grantham's test, and thus our proposed test, when $n \equiv 1 \pmod 4$, has been found yet. Both these tests are, apart from their random choice of the parameters, refinements of the test by Baillie and Wagstaff [2]. However, the latter has proved to be extremely powerful in that nobody has yet claimed the 620 prize that is offered for a composite that passes it.

Indeed, due to the existence of such strong tests, one might wonder, whether new pseudoprimality tests can be of any interest. We stress that in our case, the main motivation originally was of a different nature. It was actually the question of establishing an analogue of Euler's criterion in quadratic extensions. We believe that the result that we found (Theorem 1) is of interest by itself. Additionally, it then can be applied as an efficient pseudoprimality testing condition.

Acknowledgements

I am deeply grateful to Professor H.C. Williams for many very interesting conversations during my visit at the University of Waterloo in December 1999. It was during this time that Professor Williams raised the question about utilizing square root finding algorithms for pseudoprimality testing. His many valuable comments and his qualifying advice have helped me tremendously in writing this paper. Also I would like to thank the department of CACR at the University of Waterloo for providing a stimulating environment during my visit. Thanks are also due to Professor W. B. Müller for his continued support, and to the referee for his helpful suggestions.

References

1. Arnault, F.: Rabin-Miller primality test: Composite numbers which pass it. *Math. Comp.* **64** (209), 355-361 (1995).
2. Baillie, R., Wagstaff, S., Jr.: Lucas pseudoprimes. *Math. Comp.* **35**, 1391-1417 (1980).
3. Bleichenbacher, D.: *Efficiency and Security of Cryptosystems based on Number Theory*. Dissertation ETH Zürich (1996).
4. Carmichael, R.D.: On Sequences of Integers Defined by Recurrence Relations. *Quart. J. Pure Appl. Math.* Vol. 48, 343-372 (1920).
5. Grantham, J.: A Probable Prime Test with High Confidence. *J. Number Theory* **72**, 32-47 (1998).
6. Grantham, J.: Frobenius Pseudoprimes, preprint (1998).
7. Jaeschke, G.: On strong pseudoprimes to several bases. *Math. Comp.* **61**, 915-926 (1993).
8. Koblitz, N.: *A Course in Number Theory and Cryptography*. Springer-Verlag (1994).
9. Menezes, A., Oorschot, P., Vanstone, S.: *Handbook of Applied Cryptography*, CRC Press (1996).
10. Montgomery, P.: Evaluating recurrences of form $X_{m+n} = f(X_m, X_n, X_{m-n})$ via Lucas chains, preprint.
11. More, W.: The LD Probable Prime Test. In: Mullin, R.C., Mullen, G. (eds.) *Contemporary Mathematics* **225**, 185-191 (1999).
12. Müller, S.: On the Combined Fermat/Lucas Probable Prime Test. In: Walker, M. (ed.) *Cryptography and Coding*, pp. 222-235, *Proceedings of the Seventh IMA Conference on Cryptography and Coding*, December 20-22, 1999, Royal Agricultural College, Cirencester, UK, LNCS 1746, Springer - Verlag (1999).
13. Müller, W.B., Oswald, A.: Dickson pseudoprimes and primality testing. In: Davies, D.W. (ed.) *Advances in Cryptology - EUROCRYPT'91*, 512-516. Lecture Notes in Computer Science, Vol. 547. Berlin Heidelberg New York, Springer (1991).
14. Ribenboim, P.: *The New Book of Prime Number Records*. Berlin, Springer (1996).
15. Somer, L.: Periodicity Properties of k th Order Linear Recurrences with Irreducible Characteristic Polynomial Over a Finite Field, In: *Finite Fields, Coding Theory and Advances in Communications and Computing*, Gary L. Mullen and Peter Jau-Shyong Shiue (eds.), Marcel Dekker Inc., 195-207 (1993).
16. Somer, L.: On Lucas d -Pseudoprimes. In: Bergum, G., Philippou, A., Horadam, A. (eds.) *Applications of Fibonacci Numbers*, Vol. 7, Kluwer, 369-375 (1998).
17. Joye M., Quisquater, J.J.: Efficient computation of full Lucas sequences, *IEE Electronics Letters*, 32.6, 537-538 (1996).
18. Williams, H.C: *Édouard Lucas and primality Testing*, John Wiley & Sons (1998).

Improving Group Law Algorithms for Jacobians of Hyperelliptic Curves

Koh-ichi Nagao

Department of Basic Science, Faculty of Engineering
Kanto-Gakuin University
4834 Mutsuura cho, Kanazawa Ku, Yokohama City 236-8501, Japan
nagao@kanto-gakuin.ac.jp

Abstract. In this paper, we propose three ideas to speed up the computation of the group operation in the Jacobian of a hyperelliptic curve:
1. Division of polynomials without inversions in the base field, and an extended gcd algorithm which uses only one inversion in the base field.
2. The omission of superfluous calculations in the reduction part.
3. Expressing points on the Jacobian in a slightly different form.

1 Introduction

Using the Jacobian of a hyperelliptic curve defined over a finite field for applications to cryptology was first proposed by Koblitz [6]. Practical algorithms for the group operation in such Jacobians have been proposed by Cantor [3] and Koblitz [7], [8]. Actual computations were done by Sakai, Ishizuka, Sakurai [12], and others. In the case of an elliptic curve, an algorithm for speeding up the group operation computations was proposed by Miyaji, Ono and Cohen [10].

The present paper discusses three ideas to speed up the computation of the group operation in the Jacobian of a hyperelliptic curve. The improvement is confirmed by counting the number of operations in the base field used.

The time needed for a multiplications and b inversions in the base field we denote by $aM+bI$ (additions in the base field are ignored). Although it is believed that $I \doteq (10 + \epsilon)M$, for actual processors it is reported that $I \doteq (20 + \epsilon)M$ (Dec Alpha processor) and $I \doteq (30 + \epsilon)M$ (Intel Pentium processor); see Futa [5].

2 Cantor's Algorithm

In this section, we review the computation of the group operation in the Jacobian of a hyperelliptic curve as proposed by Cantor [3].

Let K be a perfect field with $\text{char}(K) \neq 2$, and suppose C is a hyperelliptic curve of genus g , defined over K , given by an equation $y^2 = F(x)$ for some $F \in K[x]$ of degree $2g+1$. Its Jacobian (more precisely, the K -rational points of it) can be regarded as the set

$$J(C)(K) := \{(a, b) \in \mathbb{A}^2(K[x]) \mid a \text{ monic}, \deg b < \deg a \leq g, F - b^2 \equiv 0 \pmod{a}\}.$$

The group structure on $J(C)(K)$ has $(1, 0)$ as unit element, and the inverse of (a, b) is $(a, -b)$. Let $P_i = (a_i(x), b_i(x)) \in J(C)(K)$ be two points. Cantor's algorithm for computing $P_1 + P_2 = P_3 = (a_3(x), b_3(x))$ requires the next two parts.

(Ideal Composition Part)

- 1-1. Take e_1, e_2, d_1 such that $d_1 = \gcd(a_1, a_2)$ (monic) and $e_1 a_1 + e_2 a_2 = d_1$.
- 1-2. Take c_1, c_2, d such that $d = \gcd(d_1, b_1 + b_2)$ (monic) and $c_1 d_1 + c_2 (b_1 + b_2) = d$.
- 1-3. Put $s_1 = c_1 e_1, s_2 = c_2 e_2, s_3 = c_2$, then we have $d = s_1 a_1 + s_2 a_2 + s_3 (b_1 + b_2)$.
- 1-4. Put $a = a_1 a_2 / d^2$ and $b = (s_1 a_1 b_2 + s_2 a_2 b_1 + s_3 (b_1 b_2 + F)) / d \bmod a$.

(Reduction Part)

- 2-1. Put $a' = (F - b^2) / a$ and $b' = (-b \bmod a')$.
- 2-2. If $\deg a' > g$ then $a \leftarrow \text{monic}(a')$, $b \leftarrow b'$, goto 2-1.
- 2-3. Put $a_3 \leftarrow \text{monic}(a')$, $b_3 \leftarrow b'$ and output $(a_3(x), b_3(x))$.

3 Definition of Type I and Type II

Definition 1 (Type I). *The addition $P_1 + P_2$, (where $P_i = (a_i(x), b_i(x))$ ($i = 1, 2$)), is called an addition of type I, if the conditions $\gcd(a_1, a_2) = 1$ and $\deg(a_1) = \deg(a_2) = g$ are satisfied.*

In this case, the algorithm of Cantor can be simplified as follows.

The Simplified Addition Algorithm for Type I

- I-1. Take e_1, e_2 such that $e_1 a_1 + e_2 a_2 = 1$.
- I-2. Put $a = a_1 a_2$ and $b = e_1 a_1 b_2 + e_2 a_2 b_1 \bmod a$.
- I-3. Put $a_3 = (b^2 - F) / a$.
- I-4. $a_3 \leftarrow \text{monic}(a_3)$ and $b_3 = -b \bmod a_3$.
- I-5. If $\deg a_3 > g$ then $a \leftarrow a_3$, $b \leftarrow b_3$, goto I-3, else output (a_3, b_3) .

Definition 2 (Type II). *The duplication $2P_1$, (where $P_1 = (a_1(x), b_1(x))$), is called a duplication of type II, if the conditions $\gcd(a_1, b_1) = 1$ and $\deg(a_1) = g$ are satisfied.*

Also in this case, the algorithm of Cantor can be simplified:

The Simplified Duplication Algorithm for Type II

- II-1. Take e_1, e_2 satisfying $e_1 a_1 + 2e_2 b_1 = 1$.
- II-2. Put $a = a_1^2$ and $b = e_1 a_1 b_1 + e_2 (b_1^2 + F) \bmod a$.
- II-3. Put $a_3 = (b^2 - F) / a$.
- II-4. $a_3 \leftarrow \text{monic}(a_3)$ and $b_3 = -b \bmod a_3$.
- II-5. If $\deg a_3 > g$ then $a \leftarrow a_3$, $b \leftarrow b_3$, goto II-3, else output (a_3, b_3) .

Lemma 1. *Assume that the base field is $K = \mathbb{F}_p$. Let P_1, P_2 (resp. P_1) be (a) random point(s) of $J(C)(\mathbb{F}_p)$. The probability that the addition $P_1 + P_2$ (resp. the duplication $2P_1$) is not of type I (resp. type II) is $O(\frac{1}{p})$.*

Proof. We only prove this for duplications and type II, since the remaining case is analogous. Let $(a(x), b(x))$ be a point of $J(C)(\mathbb{F}_p)$ which does not give a duplication of type II. Since $\#J(C)(\mathbb{F}_p)$ is roughly p^g and the number of monic polynomials whose degree is less than g is p^{g-1} , it follows that the probability that $\deg(a(x)) < g$ is only $O(\frac{1}{p})$. Hence we may and will assume that $\deg(a(x)) = g$ and $\gcd(a(x), b(x)) \neq 1$. The condition $a(x)|b^2(x) - F(x)$ now implies that a monic $h(x) \in \mathbb{F}_p[x]$ exists such that $h(x)|a(x)$, $0 < \deg(h(x)) < g$, and $h(x)|F(x)$.

Let $H(x)$ be any monic polynomial in $\mathbb{F}_p[x]$ with $0 < m := \deg(H(x)) < g$ and $H(x)|F(x)$. Then

$$\#\{a(x) \in \mathbb{F}_p[x] \mid a(x) \text{ monic, } \deg(a(x)) = g, H(x)|a(x)\} = p^{g-m},$$

$$\#\{(a(x), b(x)) \in J(C)(\mathbb{F}_p) \mid \deg a(x) = g, H(x)|a(x)\} \leq 2^g p^{g-m}, \text{ and}$$

$$\frac{\#\{(a(x), b(x)) \in J(C)(\mathbb{F}_p) \mid H(x)|a(x)\}}{\#J(C)(\mathbb{F}_p)} \leq O\left(\frac{1}{p}\right).$$

The number of possibilities for $H(x)$ is finite and independent of p , so we obtain the desired result.

The lemma shows that almost all additions and duplications satisfy the condition of type I and type II, respectively. Therefore, we will only discuss these cases in the following.

Lemma 2. *In the computation of an addition or duplication of type I and type II respectively, the number of loops in the simplified algorithm is at most $[\frac{g+1}{2}]$, where $[r]$ denotes the largest integer $\leq r$.*

Proof. Initially one has $\deg(a_3) \leq 2g - 2$. If $\deg(a_3) \geq g + 2$ then $\deg(a_3)$ decreases by at least two in every cycle. In case $\deg(a_3) = g + 1$ only one more cycle is necessary. This implies the result.

4 Division of Polynomials without Using Inversions

Consider the division with remainder $f(x) = g(x) \cdot s(x) + r(x)$ of a polynomial $f(x)$ by a monic polynomial $g(x)$. Here $s(x)$ and $r(x)$ are easily computed using synthetic division. However in case $g(x)$ is not monic this computation requires inversion in the base field, which is time consuming.

We propose two algorithms for the division of polynomials, which compute polynomials $S(x)$ and $R(x)$ and a constant D such that

$$f(x) = g(x) \cdot \frac{S(x)}{D} + \frac{R(x)}{D}.$$

These algorithms do not involve any inversions. Write

$$f(x) = \sum_{i=0}^n a_i x^i, \quad g(x) = \sum_{i=0}^m b_i x^i, \quad S(x) = \sum_{i=0}^{n-m} c_i x^i, \quad R(x) = \sum_{i=0}^{m-1} d_i x^i.$$

We assume $n = \deg f(x) \geq \deg g(x) = m$.

Algorithm 1 (Faster when $n \gg m$)

1. $\beta \leftarrow b_m$, make a table of the β^i ($i = 0, \dots, n$).
2. $W_i \leftarrow a_i \beta^{n-i}$ ($i = 0, \dots, n-1$).
3. Make a table of the $\beta^{i-1} b_{m-i}$ ($i = 1, \dots, m$).
4. Loop: j moves from 1 to $n-m+1$.
 - 4-1. Put $r = c_{n-m+1-j} \leftarrow W_{n+1-j}$
 - 4-2. Put $W_{n+1-i-j} = W_{n+1-i-j} - r \beta^{i-1} b_{m-i}$ ($i = 1, \dots, m$).
(End of loop)
 5. Take $d_i = W_i \beta^i$ ($i = 0, \dots, m-1$).
 6. Put $c_i \leftarrow c_i \beta^{m-1+i}$ ($i = 0, \dots, n-m$).
 7. Take $D = \beta^n$.

Algorithm 2 (Faster when $n \doteq m$)

1. $\beta \leftarrow b_m$, $W_i \leftarrow a_i$ ($i = 0, \dots, n-1$).
2. Make a table of the β^i ($i = 1, \dots, n-m+1$).
3. Loop: j moves from 1 to $n-m+1$.
 - 3-1. Put $r = c_{n-m+1-j} \leftarrow W_{n+1-j}$
 - 3-2. Put $W_{n+1-i-j} = \beta W_{n+1-i-j} - r b_{m-i}$ ($i = 1, \dots, m$).
 - 3-3. If $n-m-j \geq 0$ then $W_{n-m-j} = W_{n-m-j} \beta^j$.
(End of loop)
 4. Take $d_i = W_i$ ($i = 0, \dots, m-1$).
 5. Put $c_i \leftarrow c_i \beta^i$ ($i = 0, \dots, n-m$).
 6. Take $D = \beta^{n-m+1}$.

5 Extended Gcd with Only One Inversion

In general the Euclidean algorithm (cf. [4]) is used for an extended gcd calculation, which means computing $e_1(x)$, $e_2(x)$, and a monic $\gcd(f(x), g(x))$ such that $e_1(x)f(x) + e_2(x)g(x) = \gcd(f(x), g(x))$ for given polynomials $f(x)$, $g(x)$. We will apply our division of polynomials to this. Assume $\deg f(x) \geq \deg g(x)$.

Improved Euclidean Algorithm

1. $f_0(x) \leftarrow f(x)$, $g_0(x) \leftarrow g(x)$, $i = 0$.
2. Apply division with remainder to obtain $S_i(x)$, $R_i(x)$ and a constant D_i such that $f_i(x) = g_i(x) \cdot S_i(x)/D_i + R_i(x)/D_i$.
3. If $R_i(x) = 0$ then $i \leftarrow i-1$, goto 7.
4. Put $\begin{pmatrix} f_{i+1} \\ g_{i+1} \end{pmatrix} \leftarrow \begin{pmatrix} g_i \\ R_i \end{pmatrix}$, $M_i \leftarrow \begin{pmatrix} 0 & 1 \\ D_i & -S_i \end{pmatrix}$; now $\begin{pmatrix} f_{i+1} \\ g_{i+1} \end{pmatrix} = M_i \cdot \dots \cdot M_0 \begin{pmatrix} f \\ g \end{pmatrix}$.
5. If $\deg g_{i+1}(x) = 0$ then goto 7.
6. $i \leftarrow i+1$, goto 2 .
7. If $i \geq 0$ then compute $e_1(x)$, $e_2(x)$ by $\begin{pmatrix} \times & \times \\ e_1(x) & e_2(x) \end{pmatrix} = M_i \cdot M_{i-1} \cdots M_0$
(and if $i = -1$ then $e_1(x) \leftarrow 0$, $e_2(x) \leftarrow 1$).
8. Put α as leading coefficient of $g_{i+1}(x)$.
9. Put $\gcd(f(x), g(x)) \leftarrow \alpha^{-1} g_{i+1}(x)$, $e_1(x) \leftarrow \alpha^{-1} e_1(x)$, $e_2(x) \leftarrow \alpha^{-1} e_2(x)$.

The algorithm above computes only once inversion (α^{-1}). Hence it is faster than the original Euclidean algorithm, if inversion in the base field is much slower than multiplication. This is the case, for example, if the base field is \mathbb{F}_p for a large prime p . Since the extended gcd algorithm is used in the algorithms for computing in the group $J(C)(\mathbb{F}_p)$, these will perform faster as well.

6 Improvement of the Reduction Part

In the computation of the reduction part, the division $a' = (b^2 - F)/a$ turns out to be a bottleneck step. The fact that this is a division with remainder 0, allows one to omit some calculations of low degree terms here. More precisely, put $n = \deg a$, then the following holds.

1. In the computation of b^2 , only the terms with degree $\geq n$ are needed.
 2. To compute $(b^2 - F)/a$, only terms of $b^2 - F$ with degree $\geq n$ are needed.
- These observations allow one to make the we computation of $(b^2 - F)/a$ three to four times as fast.

7 Expressing Points on $J(C)(K)$ Using a Denominator

In section 5, we propose a improved extended gcd algorithm which computes $e_1(x)f(x) + e_2(x)g(x) = \gcd(f(x), g(x))$. Since we required the gcd to be monic, this algorithm needs one inversion in the base field. If we allow the gcd to be a non-monic polynomial, this inversion is not needed anymore. In this section, we attempt to use an extended gcd algorithm which outputs as gcd a possibly non-monic polynomial, to the computation of the group law in $J(C)(K)$. For this purpose, the second coordinate of a point in $J(C)(K)$ will be expressed as $b(x) = B(x)/l$ for some $l \in K^\times$.

The numerical experiments described in the next section suggest that if the genus g is an even number, this leads to an improvement. However, if the genus g is odd, the performance is not better (if g is odd, then a' obtained in the last cycle of the reduction part must be monic, and some calculations are omitted.)

We now present the algorithms for the group operations of type I and type II, improved using sections 5, 6, and 7. Let P_1, P_2 be points in $J(C)(K)$ written in the form $P_i = (a_i(x), B_i(x)/l_i)$ for $i = 1, 2$.

Algorithm for Type I (Computation of $P_1 + P_2$)

I-1. Take E_1, E_2, d as $E_1a_1 + E_2a_2 = d$. (Remark that $d \in K^\times$.)

I-2. Put $a = a_1a_2$, $B = E_1a_1B_2l_1 + E_2a_2B_1l_2 \bmod a$, $l = l_1l_2d$.

Here B is computed using the following steps:

I-2-1. Compute E_1a_1 .

I-2-2. Compute $(E_1a_1)(B_2l_1)$.

I-2-3. Compute $(d - E_1a_1)(B_1l_2)$ which equals $E_2a_2B_1l_2$.

I-2-4. Put $B = E_1a_1B_2l_1 + E_2a_2B_1l_2 \bmod a$.

I-3 (Reduction Step)

I-3-1 Compute l^2 .

I-3-2 Compute the coefficients of l^2F whose degree $\geq \deg a$.

I-3-3 Compute the coefficients of B^2 whose degree $\geq \deg a$.

I-3-4 Put $a' = (l^2F - B^2)/a$.

I-3-5 Take $a' \leftarrow \text{monic}(a')$.

I-3-6 Put $B' = -B \bmod a'$.

I-3-7 If $\deg a' > g$ then $a \leftarrow a'$, $B \leftarrow B'$ and goto I-3-2, else output $(a(x), B(x)/l)$.

Algorithm of Type II (Computation of $2P_1$)

II-1-1 Compute $(l_1 a_1)$, $2B_1 (= B_1 + B_1)$.

II-1-2. Take E_1, E_2, d as $E_1(l_1 a_1) + E_2(2b_1) = d$. (Remark that $d \in K^\times$.)

II-2. Put $a = a_1^2$, $B = E_1 a_1 B_1 l_1 + E_2 B_1^2 + E_2 l_1^2 F \bmod a$, $l = l_1 d$.

Here B is computed using the following steps:

II-2-1. Compute $E_2 B_1$.

II-2-2. Compute $B_1(d - E_2 B_1)$ which equals to $E_1 a_1 B_1 l_1 + E_2 B_1^2$.

II-2-3. Compute l_1^2 and $E_2 l_1^2$.

II-2-4. Put $B = E_1 a_1 B_1 l_1 + E_2 B_1^2 + E_2 l_1^2 F \bmod a$

II-3 (Reduction Step)

II-3-1 Compute l^2 .

II-3-2 Compute the coefficients of l^2F whose degree $\geq \deg a$.

II-3-3 Compute the coefficients of B^2 whose degree $\geq \deg a$.

II-3-4 Put $a' = (l^2F - B^2)/a$.

II-3-5 Take $a' \leftarrow \text{monic}(a')$.

II-3-6 Put $B' = -B \bmod a'$.

II-3-7 If $\deg a' > g$ then $a \leftarrow a'$, $B \leftarrow B'$ and goto II-3-2, else output

$(a(x), B(x)/l)$.

8 Numerical Experiment

In this section, we evaluate the computational quantity $aM + bI$ for our group law algorithms on $J(C)(\mathbb{F}_p)$. For this purpose, we make low level functions for inversion and multiplication modulo p , and count how often these functions are called. By convention, we do not count a trivial operation, for example, the cases $a \times 0 = 0$, $2 \times a = (a+a) \bmod p$, $a/1 = a$ etc. All further programs for operations on polynomials, the extended gcd, and the group law algorithm use only these low level functions when dealing with multiplications and inversions modulo p .

The experiments were done by taking various prime numbers $p \doteq 10^5$. The following tables show the maximal values of aM and bI over the chosen set of primes. We remark that in almost all cases the maximal values $aM + bI$, are not very different from the value for a fixed used prime p .

In the tables the following notation is used.

A: without any suggested improvements.

B: using only the improved extended gcd discussed in § 5.

C: using the improved extended gcd and the reduction part as in § 6.

D: using all three improvements discussed in this paper.

Addition of Type I
 $C : y^2 = x^{2g+1} + x^{2g} + \dots + x^2 + x + 1$

g:genus	A	B	C	D
2	70 M+ 3 I	71 M+ 2 I	52 M+ 2 I	55 M+ 1 I
3	200 M+ 4 I	204 M+ 2 I	144 M+ 2 I	154 M+ 2 I
4	386 M+ 6 I	398 M+ 3 I	286 M+ 3 I	289 M+ 2 I
5	694 M+ 7 I	717 M+ 3 I	496 M+ 3 I	510 M+ 3 I
6	1054 M+ 9 I	1091 M+ 4 I	756 M+ 4 I	759 M+ 3 I
7	1604 M+ 10 I	1658 M+ 4 I	1114 M+ 4 I	1132 M+ 4 I
8	2186 M+ 12 I	2260 M+ 5 I	1516 M+ 5 I	1519 M+ 4 I
9	3042 M+ 13 I	3139 M+ 5 I	2054 M+ 5 I	2076 M+ 5 I
10	3894 M+ 15 I	4017 M+ 6 I	2622 M+ 6 I	2625 M+ 5 I

Addition of Type I
 $C : y^2 = x^{2g+1} + A_{2g}x^{2g} + \dots + A_1x + A_0 \quad A_i:\text{random numbers}$

g:genus	A	B	C	D
2	70 M+ 3 I	71 M+ 2 I	52 M+ 2 I	56 M+ 1 I
3	200 M+ 4 I	204 M+ 2 I	144 M+ 2 I	157 M+ 2 I
4	386 M+ 6 I	398 M+ 3 I	286 M+ 3 I	292 M+ 2 I
5	694 M+ 7 I	717 M+ 3 I	496 M+ 3 I	515 M+ 3 I
6	1054 M+ 9 I	1091 M+ 4 I	756 M+ 4 I	764 M+ 3 I
7	1604 M+ 10 I	1658 M+ 4 I	1114 M+ 4 I	1139 M+ 4 I
8	2186 M+ 12 I	2260 M+ 5 I	1516 M+ 5 I	1526 M+ 4 I
9	3042 M+ 13 I	3139 M+ 5 I	2054 M+ 5 I	2085 M+ 5 I
10	3894 M+ 15 I	4017 M+ 6 I	2622 M+ 6 I	2634 M+ 5 I

Duplication of Type II
 $C : y^2 = x^{2g+1} + x^{2g} + \dots + x^2 + x + 1$

g:genus	A	B	C	D
2	66 M+ 3 I	68 M+ 2 I	49 M+ 2 I	55 M+ 1 I
3	186 M+ 4 I	192 M+ 2 I	132 M+ 2 I	146 M+ 2 I
4	359 M+ 6 I	372 M+ 3 I	260 M+ 3 I	268 M+ 2 I
5	650 M+ 7 I	673 M+ 3 I	452 M+ 3 I	472 M+ 3 I
6	989 M+ 9 I	1025 M+ 4 I	690 M+ 4 I	700 M+ 3 I
7	1514 M+ 10 I	1566 M+ 4 I	1022 M+ 4 I	1048 M+ 4 I
8	2067 M+ 12 I	2138 M+ 5 I	1394 M+ 5 I	1406 M+ 4 I
9	2890 M+ 13 I	2983 M+ 5 I	1898 M+ 5 I	1930 M+ 5 I
10	3705 M+ 15 I	3823 M+ 6 I	2428 M+ 6 I	2442 M+ 5 I

Duplication of Type II

$C : y^2 = x^{2g+1} + A_{2g}x^{2g} + \dots + A_1x + A_0$ for random numbers A_i

g:genus	A	B	C	D
2	76 M+ 3 I	78 M+ 2 I	59 M+ 2 I	66 M+ 1 I
3	207 M+ 4 I	213 M+ 2 I	153 M+ 2 I	170 M+ 2 I
4	395 M+ 6 I	408 M+ 3 I	296 M+ 3 I	307 M+ 2 I
5	705 M+ 7 I	728 M+ 3 I	507 M+ 3 I	532 M+ 3 I
6	1067 M+ 9 I	1103 M+ 4 I	768 M+ 4 I	783 M+ 3 I
7	1619 M+ 10 I	1671 M+ 4 I	1127 M+ 4 I	1160 M+ 4 I
8	2203 M+ 12 I	2274 M+ 5 I	1530 M+ 5 I	1549 M+ 4 I
9	3061 M+ 13 I	3154 M+ 5 I	2069 M+ 5 I	2110 M+ 5 I
10	3915 M+ 15 I	4033 M+ 6 I	2638 M+ 6 I	2661 M+ 5 I

Appendix

Dr. Y. Futa of Matsushita Electronics suggested that the improved extended gcd algorithm in section 5 may be applied to the computation of inversions in \mathbb{F}_{p^n} . Let $f(x) = x^n + ax^{n-1} + \dots + b \in \mathbb{F}_p[x]$ be a monic irreducible polynomial. Then an element of $\mathbb{F}_{p^n} = \mathbb{F}_p[x]/(f(x))$ is represented by a polynomial $g(x)$ with $\deg g(x) < n$. Computing the extended gcd $e_1(x)f(x) + e_2(x)g(x) = 1$ yields the inverse element $g(x)^{-1} = e_2(x)$. Futa also suggested that the time needed for this inversion is essentially less than that for the extended gcd, because the computation of $e_1(x)$ can be omitted.

Futa and I estimate that the above idea uses $(\frac{7}{2}n^2 - \frac{3}{2}n - 3)M + I$. In the special case where $f(x)$ is of the form $x^n - w$, even $(\frac{7}{2}n^2 - \frac{3}{2}n - 4)M + I$ suffices. When n is more than 6 or 7, this value may be smaller than the value used in Baily and Paar's method [2], which is known as the fastest inversion method. It should be noted that our method does not need the condition on p called OEF [1].

Postscript

Further actual computation of the Jacobian of a hyper elliptic curve defined over \mathbb{F}_{2^n} is done by Tamura and Sakurai [14]. The similar computation of extended gcd is also done by Lim and Hwang [9].

Acknowledgement

The author would like to thank Dr. Shigenori Uchiyama at NTT and Dr. Yuichi Futa at Matsushita Electronics for their fruitful discussions and comments. Moreover, the author would like to thank the referee and the editor, who pointed out many English mistakes.

References

1. D. V. Baily, C. Paar: Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms. CRYPTO'98, LNCS **1462** (1998) 472–485
2. D. V. Baily, C. Paar: Elliptic Curve Cryptosystems over Large Characteristic Extension Fields. preprint, (1999)
3. D. Cantor: Computing in the Jacobian of a Hyperelliptic curve. Math Comp. **48** (1987) 95–101
4. H. Cohen: A Course in Computational Algebraic Number Theory. GTM **138**, Springer (1993)
5. Y. Futa, A Miyaji: An efficient Construction of Elliptic curves over OEF fields. Talks in Kyoto Rigar Royal Hotel (1999)
6. N. Koblitz: Elliptic Cryptosystems. Math Comp. **48** (1987) 203–209
7. N. Koblitz: Hyperelliptic Cryptosystems. J. Cryptology **1** (1989) 139150
8. N. Koblitz: Algebraic Aspects of Cryptography, Algorithms and Computations in Math., **vol. 3**, Springer (1998)
9. G. H. Lim and H. S. Hwang: Fast Implementation of Elliptic Curve Arithmetic in $\text{GF}(p^n)$, PKC 2000, LNCS **1751** (2000) 405–421
10. A. Miyaji, T. Ono, H. Cohen: Efficient Elliptic Curve Exponentiation. SCIS'98. (Hamanakako, Jan., 1998)
11. A.J. Menezes, Y. Wu, R..J. Zuccherato: An elementary Introduction to Hyperelliptic Curves. Appendix to Algebraic Aspects of Cryptography, (au. N. Koblitz), Springer, (1998)
12. Y. Sakai, H. Ishizuka, K. Sakurai: Construction and Implementation of Secure Hyperelliptic Cryptosystems, SCIS'98, (Hamanakako, Jan.,1998)
13. N. Smart: On the Perfomance of Hyperelliptic Cryptosystems. EuroCrypto'99, LNCS **1592** (1999) 165–175
14. T. Tamura and K. Sakurai: Genus Dependency of Hardware Implementation of Hyperelliptic Curve cryptosystems. SCIS 2000, (Okinawa, Jan., 2000)

Central Values of Artin L -Functions for Quaternion Fields

Sami Omar

Université Bordeaux I, Laboratoire A2X
351 cours de la Libération, 33 405 Talence, France
omar@math.u-bordeaux.fr

Abstract. Using Weil explicit Formulas, we show how to compute the multiplicity n_χ of a zero at the point $\frac{1}{2}$ of the Artin L -functions associated to a character χ of Degree 2 in quaternion fields of degree 8. We prove in several examples that $n_\chi = 0$ when $W(\chi) = 1$ and $n_\chi = 1$ when $W(\chi) = -1$.

1 Basic Properties of Artin L -Functions and Conjectures

Let N/K be a Galois extension of a number field with a group $G = Gal(N/K)$ and let (ρ, V) be a representation of G and χ denotes its character, then the Artin L -function attached to χ is defined by:

$$L(N/K, \chi, s) = \prod_{\mathfrak{p} \nmid d_K} \frac{1}{\det(1 - \varphi_{\mathfrak{P}} N(\mathfrak{p})^{-s})};$$

where the product is over unramified primes of K and $\varphi_{\mathfrak{P}}$ is the Frobenius automorphism. The Artin L -function converges uniformly in the half-plane $\Re(s) > 1 + \delta$ ($\delta > 0$) and defines an analytic function on the half plane $\Re(s) > 1$. From basic properties of Artin L -functions we have the identity:

Theorem 1.

We have:

$$\zeta_N(s) = \zeta_K(s) \prod_{\chi \neq 1} L(N/K, \chi, s)^{\chi(1)},$$

where χ varies over the non trivial irreducible characters of G and $\chi(1)$ appear as the unique positive integers coefficients in the decomposition of the regular representation reg_G of G (cf. [1]): $\text{reg}_G = \sum \chi(1) \chi$.

In order to obtain an L -function with a functional equation, it is necessary to introduce Euler factors at infinite primes of K . Let us define:

$$\Lambda(N/K, \chi, s) = c(N/K, \chi)^{\frac{s}{2}} L_\infty(N/K, \chi, s) L(N/K, \chi, s),$$

where

$$c(N/K, \chi) = |d_K|^{\chi(1)} N(\mathfrak{f}(N/K, \chi))$$

and

$$L_\infty = \prod_{\mathfrak{p} \mid \infty} L_{\mathfrak{p}}(N/K, \chi, s).$$

For every infinite place \mathfrak{p} of K , we put:

$$L_{\mathfrak{p}}(N/K, \chi, s) = \begin{cases} L_{\mathbb{C}}(s)^{\chi(1)} & \text{if } \mathfrak{p} \text{ is complex,} \\ L_{\mathbb{R}}(s)^{n^+} L_{\mathbb{R}}(s+1)^{n^-} & \text{if } \mathfrak{p} \text{ is real.} \end{cases}$$

where

$$L_{\mathbb{C}}(s) = 2(2\pi)^{-s} \Gamma(s), \quad L_{\mathbb{R}}(s) = \pi^{-\frac{s}{2}} \Gamma\left(\frac{s}{2}\right)$$

and

$$n^+ = \frac{\chi(1) + \chi(\varphi_{\mathfrak{P}})}{2}, \quad n^- = \frac{\chi(1) - \chi(\varphi_{\mathfrak{P}})}{2}.$$

Using the Brauer induction theorem, we have the following result:

Theorem 2.

The completed Artin function $\Lambda(N/K, \chi, s)$ has a meromorphic continuation to the complex plane and satisfies:

$$\Lambda(N/K, \chi, s) = W(\chi) \Lambda(N/K, \bar{\chi}, 1-s)$$

where $W(\chi)$ is of absolute value 1.

The reader is referred to [2] for a good introduction to the Artin L -functions and character theory.

Artin's conjecture asserts that for every irreducible character $\chi \neq 1$, the Artin L -function $L(N/K, \chi, s)$ has an analytic continuation. This means that the quotient $\frac{\zeta_N}{\zeta_K}$ is entire. Actually Aramata and Brauer proved the following theorem (cf. [3]):

Theorem 3.

If N/K is a Galois extension then the quotient $\frac{\zeta_N}{\zeta_K}$ is entire.

Now we restrict our attention to the multiplicity of a zero of an Artin L -function. In this direction Stark proved the following result (cf. [4]):

Theorem 4.

Let us denote $n_\chi(s_0) = \text{ord}_{s=s_0} L(N/K, \chi, s)$ and $r = \text{ord}_{s=s_0} \zeta_N(s)$, then we have the inequality:

$$\sum_{\chi \text{ irreducible}} n_\chi(s_0)^2 \leq r^2.$$

In the particular case where $s_0 = \frac{1}{2}$, we denote $n_\chi = n_\chi(\frac{1}{2})$. If we assume the Generalized Riemann Hypothesis (GRH), we can show the following estimate for n_χ :

Theorem 5.

We have:

$$n_\chi \leq \frac{3}{2} \frac{\ln |d_N|}{\ln \ln |d_N|}.$$

Proof.

To prove the inequality above, we use the Weil explicit formulas applied to $\zeta_N(s)$ (cf. [5]):

Theorem 6. (Weil Explicit Formulas)

Let F satisfy the conditions (A) and (B) below and $F(0) = 1$:

(A) F is continuous and continuously differentiable everywhere except at a finite number of points a_i , where $F(x)$ and $F'(x)$ have only a discontinuity of the first kind, such that $F(a_i) = \frac{1}{2}(F(a_i + 0) + F(a_i - 0))$.

(B) There is a number $b > 0$ such that $F(x)$ and $F'(x)$ are $O(e^{-(\frac{1}{2}+b)|x|})$ as $|x| \rightarrow \infty$.

Then the Mellin transform of F :

$$\Phi(s) = \int_{-\infty}^{+\infty} F(x) e^{(s-\frac{1}{2})x} dx$$

is holomorphic in every strip $-a \leq \sigma \leq 1 + a$ where $0 < a < b$, $a < 1$, and the sum $\sum \Phi(\rho)$ running over the non trivial zeros $\rho = \beta + i\gamma$ of $\zeta_N(s)$ with $|\gamma| < T$ tends to a limit as T tends to infinity. This limit is given by the formula:

$$(1) \quad \sum_{\rho} \Phi(\rho) = \Phi(0) + \Phi(1) - 2 \sum_{\mathfrak{p}, m} \frac{\ln(N(\mathfrak{p}))}{N(\mathfrak{p})^{\frac{m}{2}}} F(m \ln(N(\mathfrak{p}))) \\ + \ln(|d_N|) - n[\ln(2\pi) + \gamma + 2\ln(2)] - r_1 J(F) + n I(F)$$

where

$$J(F) = \int_0^{+\infty} \frac{F(x)}{2 \cosh(\frac{x}{2})} dx, \quad I(F) = \int_0^{+\infty} \frac{1 - F(x)}{2 \sinh(\frac{x}{2})} dx$$

and $\gamma = 0.57721566 \dots$ is the Euler constant.

Now let us define F by:

$$F(x) = \begin{cases} 1 - |x| & \text{if } |x| \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

then we have:

Lemma 1.

The Fourier transform of F is:

$$\hat{F}(u) = \left(\frac{2 \sin(\frac{u}{2})}{u} \right)^2.$$

Let us put $F_T(x) = F(\frac{x}{T})$, then $\hat{F}_T(u) = T \hat{F}(Tu)$. If we apply theorem 6 to F_T , we obtain the inequality:

$$n_\chi T \leq 4 \int_0^T \cosh(\frac{x}{2}) dx + \ln(|d_N|) + n \int_0^T \frac{x}{2T \sinh(\frac{x}{2})} dx. \\ n_\chi T \leq e^{\frac{T}{2}} + \ln(|d_N|) + n$$

we put $T = 2 \ln \ln(|d_N|)$ then:

$$n_\chi \leq \frac{\ln(|d_N|)}{\ln \ln(|d_N|)} + \frac{n}{2 \ln \ln(|d_N|)} \leq \frac{3}{2} \frac{\ln(|d_N|)}{\ln \ln(|d_N|)}.$$

□

Now we shall recall some important conjectures on n_χ and $n_\chi(s_0)$:

Conjecture 1.

In the case of the Dirichlet L series, we have $n_\chi = 0$.

Conjecture 2.

Generally for every irreducible character χ , we have:

$$n_\chi(s_0) \ll \chi(1)$$

or even the stronger result:

$$n_\chi(s_0) \ll 1.$$

2 Quaternion Extensions

In this section we describe some properties related to the construction of quaternion fields and their associated Artin L -functions. We shall restrict ourselves to the case of a tamely ramified extension.

Definition 1.

A quaternion extension of \mathbb{Q} is a normal extension N of \mathbb{Q} with Galois group G isomorphic to the quaternion group H_8 of order 8.

The quaternion group is the unique group of order 8 having 3 cyclic subgroups of order 4.

One can write $H_8 = \langle \sigma, \tau \rangle$ with relations $\sigma^4 = 1$, $\tau^2 = \sigma^2$ and $\tau\sigma\tau^{-1} = \sigma^{-1}$, so there exists a unique irreducible character of degree 2 of H_8 verifying $\chi(1) = 2$, $\chi(\sigma^2) = -2$ and $\chi(s) = 0$ for $s \neq 1, \sigma^2$.

N contains three quadratic subfields k_1, k_2, k_3 with discriminant d_1, d_2, d_3 and a biquadratic subfield K with discriminant $d_1 d_2 d_3$. One can show that N can be written $N = K(\sqrt{M})$ for some $M \in K$ such that $\text{tr}_{K/\mathbb{Q}}(M) \equiv \pm \frac{1+d_1+d_2+d_3}{2} \pmod{4}$ (cf. [6]). The theorem below tells us under what condition a quadratic field $k = \mathbb{Q}(\sqrt{d})$ can be embedded in a quaternion field N (cf. [7]).

Theorem 7.

Let d be a squarefree integer. In order that $k = \mathbb{Q}(\sqrt{d})$ is a quadratic subfield of some quaternion field N , it is necessary and sufficient that d be positive and not congruent to $-1 \pmod{8}$.

Example 1.

$k_1 = \mathbb{Q}(\sqrt{5})$, $k_2 = \mathbb{Q}(\sqrt{21})$, $k_3 = \mathbb{Q}(\sqrt{105})$ are quadratic subfields of the quaternion extension $N = K(\sqrt{M})$ where $M = \frac{5+\sqrt{5}}{2} \cdot \frac{21+\sqrt{21}}{2}$. One can verify that the extensions $N/k_1, N/k_2, N/k_3$ are cyclic and so N is a quaternion extension of \mathbb{Q} .

Example 2.

One can check the same for $k_1 = \mathbb{Q}(\sqrt{5})$, $k_2 = \mathbb{Q}(\sqrt{41})$ and $N = K(\sqrt{M})$ where $M = \frac{5+\sqrt{5}}{2} \cdot \frac{41+5\sqrt{41}}{2}$.

In the last section we will give a table of many totally real and imaginary quaternion extensions with their quadratic subfields.

Now we restrict our attention to the Artin L -function $L(s, \chi)$ associated to the unique character χ of degree 2 of H_8 . If we write $L(s, \chi)$ in terms of Dedekind zeta functions, then by using theorem 1, we have:

Proposition 1.

We write:

$$\zeta_N(s) = \zeta_K(s)L(s, \chi)^2.$$

We know that $\frac{\zeta_N(s)}{\zeta_K(s)}$ is an entire function (theorem 3), so $L^2(s, \chi)$ is holomorphic on the whole plane. Since $L(s, \chi)$ is meromorphic (theorem 2) we deduce the following proposition:

Proposition 2.

The Artin $L(s, \chi)$ -function is entire.

We define an invariant U_N of the quaternion extension N by:

$$U_N = \begin{cases} 1 & \text{if the ring of integers } O_N \text{ of } N \text{ is a free } \mathbb{Z}[G]\text{-module,} \\ -1 & \text{otherwise.} \end{cases}$$

The Fröhlich theorem gives the general equality (cf. [7], [8]):

Theorem 8.

We have:

$$W(\chi) = U_N.$$

Let us denote:

$$\varepsilon = \begin{cases} 1 & \text{if } N \text{ is real,} \\ -1 & \text{if } N \text{ is imaginary.} \end{cases}$$

In [6], one can find an effective criterion to know if O_N is a free $\mathbb{Z}[G]$ -module or not:

Theorem 9.

We have:

O_N is a free $\mathbb{Z}[G]$ -module if and only if

$$\varepsilon \prod_{p|d_N} p \equiv \frac{1 + d_1 + d_2 + d_3}{4} \pmod{4}.$$

A look at the functional equation of $L(s, \chi)$ shows:

Theorem 10.

If $W(\chi) = 1$ then n_χ is even,
If $W(\chi) = -1$ then n_χ is odd.

Conjecture 3.

If $W(\chi) = 1$ then $n_\chi = 0$,
If $W(\chi) = -1$ then $n_\chi = 1$.

In fact, conjecture 3 gives more information on n_χ than conjecture 2 insofar as it says that n_χ is the smallest possible with respect to constraints imposed by the sign of $W(\chi)$ when χ is real-valued.

3 Computation of n_χ

In this section we give an explicit method to compute n_χ and verify conjecture 3 in many cases. If we suppose that $\zeta_K(\frac{1}{2}) \neq 0$ (conjecture 1) then

$$2n_\chi = \text{ord}_{\frac{1}{2}} \zeta_N(s).$$

In practice for a given quaternion field N , we show that $\zeta_K(\frac{1}{2}) \neq 0$ and compute $\text{ord}_{\frac{1}{2}} \zeta_N(s)$. Let us write the Weil explicit formulas for ζ_N , and let us consider Serre's function $F_y(x) = e^{-yx^2}$ ($y > 0$). The Mellin transform $\Phi(s)$ of F_y is

$$\Phi(s) = \sqrt{\frac{\pi}{y}} e^{\frac{1}{4y}(s-\frac{1}{2})^2}$$

and the Fourier transform \hat{F}_y of F_y is

$$\hat{F}_y(t) = \sqrt{\frac{\pi}{y}} e^{-\frac{1}{4y}t^2}$$

If we assume GRH for ζ_N , we have $\Phi(\rho) = \hat{F}_y(t)$ where $\rho = \frac{1}{2} + it$. For every $k \geq 1$, we denote by t_k the positive imaginary part of the k^{th} zero of the Dedekind zeta function, and n_k its multiplicity. We have the identity:

$$\begin{aligned} S(y) &= n_\chi + \sum_{k \geq 2}^{+\infty} n_k e^{-\frac{t_k^2}{4y}} \\ &= e^{\frac{1}{4y}} - \sqrt{\frac{y}{\pi}} \sum_{\mathfrak{p}, m} \frac{\ln(N(\mathfrak{p}))}{N(\mathfrak{p})^{\frac{m}{2}}} e^{-y(m \ln(N(\mathfrak{p})))^2} \\ &\quad + \sqrt{\frac{y}{4\pi}} [\ln(|d_N|) - 8[\ln(2\pi) + \gamma + 2\ln(2)] - r_1 J(F_y) + 8I(F_y)]. \end{aligned}$$

Here we have $r_1 = 0$ or $r_1 = 8$. Now we need the following theorem:

Theorem 11.

We have the inequality for every $y > 0$:

$$n_\chi \leq S(y)$$

and

$$\lim_{y \rightarrow 0} S(y) = n_\chi.$$

We should notice that the advantage of Serre's function is that the series $S(y)$ converges quickly to n_χ when $y \rightarrow 0$. In practice we prove in many quaternion fields that when $W(\chi) = 1$, we have $n_\chi \leq S(y) < 2$ for some $y > 0$ and so $n_\chi = 0$. Similarly for $W(\chi) = -1$ then one can prove the inequality $n_\chi \leq S(y) < 3$ for some $y > 0$ and so $n_\chi = 1$. Now to prove that $\zeta_K(\frac{1}{2}) \neq 0$, we apply again the explicit formulas to $\zeta_K(s)$ with the same function F_y and show as we did before that

$$\text{ord}_{\frac{1}{2}} \zeta_K(s) \leq S(y) < 1$$

for some $y > 0$ and thus we have $\text{ord}_{\frac{1}{2}} \zeta_K(s) = 0$. To compute $S(y)$, we shall compute the integrals $I(F_y)$ and $J(F_y)$ with a given precision. The series over the prime ideals v_∞ in the Weil explicit formulas is truncated to

$$v_{p_0}(y) = \sum_{p \leq p_0} \sum_{\mathfrak{p} \mid (p)} \ln(N(\mathfrak{p})) \sum_{m \ln(N(\mathfrak{p})) \leq \text{cons}} \frac{e^{-y(m \ln(N(\mathfrak{p})))^2}}{N(\mathfrak{p})^{\frac{m}{2}}}.$$

$$\text{where } \text{cons} = \sqrt{\frac{c \ln(10)}{y}}.$$

The condition $m \ln(N(\mathfrak{p})) < \text{cons}$ means that we don't take into account the terms of the series less than 10^{-c} . In practice we take $c = 30$ and p_0 less than 10^6 . The number field being defined by a polynomial $P(x)$, for every prime number p prime to the index of the field, the decomposition of the ideal (p) into a product of prime ideals of the field is given by the decomposition of $P(x)$ modulo p (cf. [9]). In the case where p divides the index, we use a stronger algorithm (see algorithm 6.2.5 in [9]). Actually in both cases, we don't need to compute explicitly the factors of $P(x)$ modulo p , we just need to compute the degree of each factor in order to compute the norm of the associated prime ideal. Since N/\mathbb{Q} is a Galois extension, then one need to compute only the degree of the first irreducible polynomial appearing in the decomposition of $P(x)$ modulo p . This allows us faster computations. In fact the experimental value of $S(y)$ is $\tilde{S}(y) \geq S(y)$ and so $n_\chi \leq \tilde{S}(y)$. One can prove the following estimate by using the prime number theorem:

Theorem 12.

If we take $\text{cons} = \infty$, then the following estimate holds:

$$|v_\infty(y) - v_{p_0}(y)| \ll_y \frac{\sqrt{p_0}}{\ln(p_0)} e^{-y \ln(p_0)^2}.$$

In the following computations, we give the reduced polynomial of the quaternion field N/\mathbb{Q} , the discriminant d_N of N , two quadratic subfields $\mathbb{Q}(\sqrt{d_1})$ and $\mathbb{Q}(\sqrt{d_2})$ of N (the third one is in fact $\mathbb{Q}(\sqrt{d_1 d_2})$), $W(\chi)$ and n_χ .

- 1) $N/\mathbb{Q}: P(x) = x^8 - x^7 - 34x^6 + 29x^5 + 361x^4 - 305x^3 - 1090x^2 + 1345x - 395$
 $d_N = 3^6 \cdot 5^6 \cdot 7^6$
quadratic subfields: $\mathbb{Q}(\sqrt{5})$ and $\mathbb{Q}(\sqrt{21})$
 $W(\chi) = 1$
 $n_\chi = 0.$

- 2) $N/\mathbb{Q}: P(x) = x^8 + 315x^6 + 34020x^4 + 1488375x^2 + 22325625$
 $d_N = 3^6 \cdot 5^6 \cdot 7^6$
quadratic subfields: $\mathbb{Q}(\sqrt{5})$ and $\mathbb{Q}(\sqrt{21})$
 $W(\chi) = -1$
 $n_\chi = 1.$

- 3) $N/\mathbb{Q}: P(x) = x^8 - 205x^6 + 13940x^4 - 378225x^2 + 3404025$
 $d_N = 5^6 \cdot 41^6$
quadratic subfields: $\mathbb{Q}(\sqrt{5})$ and $\mathbb{Q}(\sqrt{41})$
 $W(\chi) = -1$
 $n_\chi = 1.$

- 4) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 + 142x^6 - 115x^5 + 6641x^4 + 3055x^3 + 157938x^2 + 152941x + 2031361$
 $d_N = 3^4 \cdot 5^6 \cdot 41^6$
quadratic subfields: $\mathbb{Q}(\sqrt{5})$ and $\mathbb{Q}(\sqrt{41})$
 $W(\chi) = -1$
 $n_\chi = 1.$

- 5) $N/\mathbb{Q}: P(x) = x^8 - x^7 - 178x^6 - 550x^5 + 7225x^4 + 44407x^3 + 55928x^2 - 45392x + 4096$
 $d_N = 3^6 \cdot 11^6 \cdot 17^6$
quadratic subfields: $\mathbb{Q}(\sqrt{17})$ and $\mathbb{Q}(\sqrt{33})$
 $W(\chi) = 1$
 $n_\chi = 0.$

- 6) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 + 106x^6 + 381x^5 + 414x^4 - 8475x^3 + 44497x^2 + 151740x + 253168$
 $d_N = 3^6 \cdot 11^6 \cdot 17^6$
quadratic subfields: $\mathbb{Q}(\sqrt{17})$ and $\mathbb{Q}(\sqrt{33})$
 $W(\chi) = -1$
 $n_\chi = 1.$

- 7) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 - 475x^6 - 2386x^5 + 56669x^4 + 732202x^3 + 3280440x^2 + 5788174x + 2396941$
 $d_N = 37^6 \cdot 41^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{41})$
 $W(\chi) = -1$
 $n_\chi = 1.$
- 8) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 - 847x^6 - 4250x^5 + 194805x^4 + 2321042x^3 + 4218300x^2 - 28827252x - 48031623$
 $d_N = 37^6 \cdot 73^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{73})$
 $W(\chi) = -1$
 $n_\chi = 1.$
- 9) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 + 1854x^6 + 14657x^5 + 1134753x^4 + 15385779x^3 + 370857442x^2 + 2861780247x + 28470071727$
 $d_N = 3^4 \cdot 37^6 \cdot 73^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{73})$
 $W(\chi) = -1$
 $n_\chi = 1.$
- 10) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 + 1042x^6 + 8233x^5 + 284219x^4 + 4899401x^3 + 42209694x^2 + 179998937x + 404059099$
 $d_N = 3^4 \cdot 37^6 \cdot 41^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{41})$
 $W(\chi) = -1$
 $n_\chi = 1.$
- 11) $N/\mathbb{Q}: P(x) = x^8 - x^7 - 866x^6 - 2686x^5 + 197617x^4 + 1072207x^3 - 8786448x^2 - 32864208x + 159160192$
 $d_N = 7^6 \cdot 17^6 \cdot 23^6$
quadratic subfields: $\mathbb{Q}(\sqrt{17})$ and $\mathbb{Q}(\sqrt{161})$
 $W(\chi) = 1$
 $n_\chi = 0.$
- 12) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 - 1591x^6 - 7978x^5 + 718061x^4 + 8174530x^3 - 29006964x^2 - 433628432x + 235862473$
 $d_N = 37^6 \cdot 137^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{137})$
 $W(\chi) = -1$
 $n_\chi = 1.$
- 13) $N/\mathbb{Q}: P(x) = x^8 - 3x^7 + 3478x^6 + 27505x^5 + 4489397x^4 + 53881703x^3 + 2972520282x^2 + 26220344507x + 651061429207$
 $d_N = 3^4 \cdot 37^6 \cdot 137^6$
quadratic subfields: $\mathbb{Q}(\sqrt{37})$ and $\mathbb{Q}(\sqrt{137})$
 $W(\chi) = -1$
 $n_\chi = 1.$

References

- [1] J. -P. Serre, Linear representation of finite groups, Springer-Verlag, New York, 1977.
- [2] J. Martinet, Character theory and Artin L -functions, Algebraic Number Fields (Fröhlich), New York, 1977.
- [3] M. R. Murty, V. K. Murty, Non-vanishing of L -functions and Applications, Progress in Mathematics, Birkhäuser Verlag, 1997.
- [4] H. M. Stark, Some effective cases of the Brauer-Siegel theorem, Invent. Math., **23**, 1974, 135-152.
- [5] G. Poitou, Sur les petits discriminants, Séminaire Delange-Pisot-Poitou, 18e année, n **6**, 1976/77.
- [6] J. Martinet, Modules sur l'algèbre du groupe quaternionien, Ann. Sci. Ecole Norm Sup, **4**, 1971, 399-408.
- [7] J. Martinet, H_8 , Proc.Sympos, Univ. Durham, 1975, 525-538.
- [8] A. Fröhlich, Artin root numbers and normal integral bases for quaternion fields, Invent.Math., **17**, 1972, 143-166.
- [9] H. Cohen, A Course in Computational Algebraic Number Theory, Graduate text in Math. **138**, Springer-Verlag, New-York, 1993.

The Pseudoprimes up to 10^{13}

Richard G.E. Pinch

2 Eldon Road, Cheltenham, Glos GL52 6TU, U.K.
`rgep@chalcodon.demon.co.uk`

Abstract. There are 38975 Fermat pseudoprimes (base 2) up to 10^{11} , 101629 up to 10^{12} and 264239 up to 10^{13} : we describe the calculations and give some statistics. The numbers were generated by a variety of strategies, the most important being a back-tracking search for possible prime factorisations, and the computations checked by a sieving technique.

1 Introduction

A (*Fermat*) *pseudoprime* (*base 2*) is a composite number N with the property that $2^{N-1} \equiv 1 \pmod{N}$.

For background on pseudoprimes and primality tests in general we refer to Bressoud [1], Brillhart et al [2], Koblitz [4], Ribenboim [12] and [13] or Riesel [14]. Previous tables of pseudoprimes were computed by Pomerance, Selfridge and Wagstaff [11].

We have shown that there are 38975 pseudoprimes up to 10^{11} , 101629 up to 10^{12} and 264239 up to 10^{13} ; all have at most 9 prime factors. Let $P(X)$ denote the number of pseudoprimes less than X and let $P(d, X)$ denote the number with exactly d prime factors. In Table 1 we give the values of $P(X)$ and $P(d, X)$ for $d \leq 9$ and X in powers of 10 up to 10^{13} .

We began the computations described here some years ago and earlier versions have already been cited in the literature. We therefore feel it appropriate to document the techniques used. The data files are available at <ftp://ftp.dpmmms.cam.ac.uk/pub/PSP> or from the author.

The pseudoprimes were generated by a variety of strategies, the most important being a back-tracking search for possible prime factorisations, and the computations checked by a sieving technique, together with a “large prime variation”.

We also used the same methods to calculate the smallest pseudoprimes with d prime factors for d up to 16. The results are given in Table 2.

2 Some Properties of Pseudoprimes

In this section we discuss some elementary properties of pseudoprimes and the overall search strategy.

X	$d = 1$	2	3	4	5	6	7	8	9	$P(X)$
10^3	0	1	2	0	0	0	0	0	0	3
10^4	0	11	11	0	0	0	0	0	0	22
10^5	0	34	34	10	0	0	0	0	0	78
10^6	0	107	89	48	1	0	0	0	0	245
10^7	1	311	229	189	20	0	0	0	0	750
10^8	2	880	485	563	124	3	0	0	0	2057
10^9	2	2455	1105	1417	563	54	1	0	0	5597
10^{10}	2	6501	2391	3435	2133	405	13	0	0	14884
$25 \cdot 10^9$	2	9581	3146	4842	3454	786	42	0	0	21853
10^{11}	2	17207	4886	7909	6845	1966	156	4	0	38975
10^{12}	2	46080	9949	17087	19132	8196	1146	37	0	101629
10^{13}	2	123877	19843	35259	49479	29064	6306	407	2	264239

Table 1. The number of pseudoprimes with d distinct prime factors up to 10^{13} .

For any odd m we let $f(m)$ denote the multiplicative order of 2 modulo m , that is, the least power $f \geq 1$ such that $2^f \equiv 1 \pmod{m}$. If $m = \prod_i p_i^{a_i}$ then $f(m) = \text{lcm} \{f(p_i^{a_i})\}$. Clearly N is a pseudoprime if and only if $f(N)$ divides $N - 1$. Further define $w(p)$ to be the largest exponent such that $p^w \mid 2^{p-1} - 1$.

In practice it seems rare to have $w(p) > 1$ and so in the main part of the search we shall consider square-free pseudoprimes. We return to this point in section 6.

We assume throughout that we are searching for pseudoprimes less than some bound X : in our computations we took $X = 10^{13}$.

Proposition 1. *Let N be a pseudoprime less than X with exactly d prime factors $p_1 \leq \dots \leq p_d$.*

1. *For each i , $f(p_i) \mid N - 1$.*
2. *Each p_i satisfies $N \equiv p_i \pmod{f(p_i)}$ and $p_i f(p_i) < X$.*
3. *For $r < d$ put $P_r = \prod_{i=1}^r p_i$. Then $p_{r+1} < (X/P_r)^{1/(d-r)}$ and p_{r+1} is prime to $f(p_i)$ for all $i \leq r$.*

Proof. Part (1) follows immediately from the condition $2^{N-1} \equiv 1 \pmod{p_i}$.

Since $f(p_i) \mid p_i - 1$, we have $N \equiv 1 \equiv p_i \pmod{f(p_i)}$, $N \equiv p_i \pmod{p_i}$ and p_i prime to $f(p_i)$, so $p_f f(p_i) \mid N - p_i$. Further, N is not prime, so $p_i f(p_i) \leq N - p_i < X$ and (2) follows.

The p_i are in increasing order so the inequality in (3) is trivial. Since $f(N) \mid N - 1$, we have $f(N)$ prime to N ; but $f(p_i) \mid f(N)$ and $p_i \mid N$, so the remainder of (3) follows. \square

We consider three classes of pseudoprimes and adopt a different strategy for each. For pseudoprimes with a repeated prime factor we use the strategy of section 6 and for square-free pseudoprimes with a prime factor greater than $X/10^4$ we use the strategy of section 5. The remaining class of square-free pseudoprimes is the most numerous and here we apply the main strategy, consisting of a precomputation and the main search, described in the next two sections.

d	C?	factors	N
2		$11 \cdot 31$	341
3	C	$3 \cdot 11 \cdot 17$	561
4		$5 \cdot 7 \cdot 17 \cdot 19$	11305
5	C	$5 \cdot 7 \cdot 17 \cdot 19 \cdot 73$	825265
6		$5 \cdot 7 \cdot 17 \cdot 19 \cdot 37 \cdot 109$	45593065
7		$7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 31 \cdot 37$	370851481
8		$5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 37 \cdot 73 \cdot 97$	38504389105
9		$7 \cdot 11 \cdot 13 \cdot 17 \cdot 31 \cdot 41 \cdot 59 \cdot 61 \cdot 97$	7550611589521
10		$7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 41 \cdot 101 \cdot 181$	277960972890601
11		$7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 41 \cdot 47 \cdot 73 \cdot 631$	32918038719446881
12		$5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 37 \cdot 59 \cdot 67 \cdot 73 \cdot 199 \cdot 241$	1730865304568301265
13		$11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 41 \cdot 43 \cdot 61 \cdot 73 \cdot 97 \cdot 127 \cdot 151$	606395069520916762801
14		$7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 31 \cdot 37 \cdot 41 \cdot 61 \cdot 73 \cdot 97 \cdot 151 \cdot 241 \cdot 251$	59989606772480422038001
15		$11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 61 \cdot 73 \cdot 97 \cdot 113 \cdot 181 \cdot 257$	6149883077429715389052001
16		$11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 71 \cdot 73 \cdot 109 \cdot 113 \cdot 127 \cdot 151 \cdot 163$	540513705778955131306570201
17	C	$13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 61 \cdot 67 \cdot 71 \cdot 73 \cdot 97 \cdot 113 \cdot 127 \cdot 211$	35237869211718889547310642241

Table 2. The smallest pseudoprimes with d prime factors, $2 \leq d \leq 17$. C denotes a Carmichael number.

3 The Precomputation

The main strategy is considerably improved by a precomputation, which also proves to be of value in the sieving methods of section 7. Let $\mathcal{A}[F, Y]$ denote the set of primes $q \leq Y$ for which $f(q) \leq F$, and let $\mathcal{B}[X]$ denote the set of primes q for which $qf(q) \leq X$. By Proposition 1(2), the prime factors of the pseudoprimes up to X are all elements of $\mathcal{B}[X]$.

We needed to find the set $\mathcal{B}[10^{13}]$. The pairs $(p, f(p))$ were found by four overlapping methods.

A. $f \leq 10^3$. The list $\mathcal{A}[10^3, \infty]$ was obtained from the list of factors of numbers of the form $2^f - 1$ with $f \leq 10^3$ in Brillhart et al [2] (the ‘‘Cunningham tables’’).

B. $10^3 \leq f \leq 10^6$. The lists $\mathcal{A}[10^4, 10^{10}]$, $\mathcal{A}[10^5, 10^9]$ and $\mathcal{A}[10^6, 10^8]$ were computed as follows. For each value of f , put $f' = \text{lcm}\{2, f\}$ and let p range over the values $\equiv 1 \pmod{f'}$. If $2^f \equiv 1 \pmod{p}$ then test p for primality by trial division. If p is prime, it is added to the corresponding list: if not, it is a pseudoprime in its own right.

C. $p \leq 10^8$. The lists $\mathcal{A}[10^7, 10^7]$ and $\mathcal{A}[10^6, 10^8]$ were computed by letting p run over primes, determined as such by trial division; factorising $p - 1$, again by trial division; finding $f(p)$ by considering the divisors e of $p - 1$ and testing whether $2^e \equiv 1 \pmod{p}$; and extracting those p for which $f(p)$ was in the desired range.

D. $10^8 \leq p \leq 2.10^9$. The lists $\mathcal{A}[10^6, 10^8]$, $\mathcal{A}[10^5, 10^9]$ and $\mathcal{A}[5.10^3, 2.10^9]$ were computed by letting p run over numbers prime to 6; computing values of $2^f \pmod{p}$ for f in the desired range by successive doubling modulo p ; testing whether f divides $p - 1$; and then checking p for primality or pseudoprimality by trial division.

4 The Main Strategy

For the main search we assume that N is a pseudoprime less than the pre-assigned bound X and with exactly d prime factors, all distinct and less than $X/10^4$: pseudoprimes not satisfying these conditions will be dealt with in subsequent sections. We obtain all such N as lists of prime factors p_1, \dots, p_d by a back-tracking search. For suitable choices of F and Y we make use of the precomputed lists $\mathcal{A}[F, Y]$.

We produce successive lists of p_1, \dots, p_{d-1} recursively, looping at each search level over all the primes permitted by Proposition 1(3).

At search level $d - 1$ put $P = \prod_{i=1}^{d-1} p_i$ and $L = f(P) = \text{lcm}\{f_1, \dots, f_{d-1}\}$. We look for primes q such that $N = Pq$ is a pseudoprime; that is, we require $2^{Pq-1} \equiv 1 \pmod{q}$ and $2^{Pq-1} \equiv 1 \pmod{P}$. The first condition is equivalent to $Pq \equiv 1 \pmod{f(q)}$ and the second to $Pq \equiv 1 \pmod{L}$. But $q \equiv 1 \pmod{f(q)}$, so we require $f(q) \mid P-1$ and $Pq \equiv 1 \pmod{L}$. We consider the possible q with $f(q) = f$ in two ways, making use of a suitable precomputed $\mathcal{A}[F, Y]$.

For every factor f of $P - 1$ which satisfies $f \leq F$, we let q run over the primes from $\mathcal{A}[F, Y]$ for which $q \leq X/P$ and $f(q) = f$. If $Pq \equiv 1 \pmod{L}$ then $N = Pq$ is a pseudoprime.

For factors f of $P - 1$ with $f > F$, or for values of q greater than X/P , we let q run over the integers satisfying $P \equiv 1 \pmod{f}$ and $f(q) = f$. These conditions imply that $Pq \equiv 1 \pmod{f}$ and $Pq \equiv 1 \pmod{L}$, so it is sufficient to run over the q satisfying $Pq \equiv 1 \pmod{\text{lcm}\{f, L\}}$. If $2^f \equiv 1 \pmod{q}$ and q is prime then $N = Pq$ is a pseudoprime (and if q is composite then it is itself a pseudoprime).

We observe that small f are likely to occur often as factors of the $P - 1$ and so the precomputation of the $\mathcal{A}[F, Y]$ gives a considerable saving. We note that Cipolla [3] used the factorisation of $2^P - 1$ in an early computation of pseudoprimes.

Testing candidates for p_i for primality is required at every stage of the calculation. We found that using a table of primes up to a suitable limit produced a considerable saving in time.

Applying the main search with $X = 10^{13}$ we used $\mathcal{A}[10^4, 10^9] \cap \mathcal{B}[10^{13}]$ as the auxiliary list of primes. There were 64575 primes in this list.

5 Pseudoprimes with Large Prime Factor

Suppose that $N = Pg$ is a square-free pseudoprime less than X with a prime factor q greater than $X/10^4$; so we are assuming that $P < 10^4$. We have $f(q) \mid P - 1$, so $f(q) < 10^4$, that is $q \mid 2^f - 1$ for some $f < 10^4$. We use the lists $\mathcal{A}[10^3, \infty]$ and $\mathcal{A}[10^4, 10^{10}]$ of primes dividing such numbers produced by methods A and B of section 3. For each q in this list, and for each $P < X/q$ with $P \equiv 1 \pmod{f(q)}$, we test whether $2^{Pq} \equiv 2 \pmod{P}$: if so, then $N = Pg$ is a pseudoprime. There are 277 pseudoprimes up to 10^{13} with a prime factor greater than 10^9 : the 9 which are less than 10^{12} are given in Table 3.

N	factors
260907275113	$89 \cdot 2931542417$
470968083601	$461 \cdot 1021622741$
542620603069	$409 \cdot 1326700741$
608041244701	$11 \cdot 41 \cdot 1348206751$
688388773637	$269 \cdot 2559066073$
710663629201	$601 \cdot 1182468601$
733007751851	$83 \cdot 8831418697$
809041003843	$499 \cdot 1621324657$
934155386445	$3 \cdot 5 \cdot 29 \cdot 2147483647$

Table 3. The 9 pseudoprimes less than 10^{12} with a prime factor greater than 10^9

6 Pseudoprimes with a Repeated Prime Factor

Recall that $w(p)$ is the largest exponent such that $p^w \mid 2^{p-1} - 1$.

Proposition 2. *Let p be an odd prime and put $f = f(p)$, $w = w(p)$. If $a \leq w$ then $f(p^a) = f$: if $a \geq w$ then $f(p^a) = p^{a-w}f$.*

Proof. The multiplicative group modulo p^a is cyclic of order $p^{a-1}(p - 1)$, and reduction modulo p maps this group onto the multiplicative group modulo p ,

which is cyclic of order $p - 1$. We conclude that $f \mid f(p^a)$, and further that the quotient $f(p^a)/f$ is a power of p .

Clearly if $a \leq w$ then $2^{p^a} = (2^p)^{p^{a-1}} \equiv 2^{p^{a-1}} \equiv \dots \equiv 2 \pmod{p^a}$, so $f(p^a) = f$. We now claim that for $a \geq w$,

$$2^{p^{a-w}f} = 1 + p^a X_a$$

where X_a is an integer with $X_a \equiv X_w \pmod{p}$. Since X_w is prime to p by definition of w , X_a is prime to p . It follows immediately that the power of p dividing $f(p^a)$ is p^{a-w} and this will complete the proof of the Proposition.

We proceed by induction. The case $a = w$ is immediate. Suppose now that $a \geq w$ and

$$2^{p^{a-w}f} = 1 + p^a X_a$$

with X_a an integer, $X_a \equiv X_w \pmod{p}$. We have

$$2^{p^{a+1-w}f} = (1 + p^a X_a)^p = 1 + p^{a+1} X_a + R$$

where R denotes the sum of the remaining terms of the binomial expansion. Since p divides the binomial coefficient $\binom{p}{r}$ for $1 \leq r \leq p-1$, we see that the power of p dividing the term $\binom{p}{r}(p^a X_a)^r$ in R is at least p^{1+ar} for $1 \leq r \leq p-1$ and at least p^{ap} for the final term. Now $p \geq 3$, so every term in R is divisible by p^{1+2a} . Now $a \geq w \geq 1$, so $1 + 2a \geq a + 2$, and $p^{a+2} \mid R$; hence

$$2^{p^{a+1-w}f} = 1 + p^{a+1} X_{a+1}$$

where $X_{a+1} \equiv X_a \pmod{p}$. This completes the induction step and the claim is proved. \square

We note that if p^a divides a pseudoprime N then $f(p^a)$ must divide $N - 1$ and so be prime to p . Hence we must have $w(p) \geq a$ and consequently p^a is itself a pseudoprime.

Suppose now that N is a pseudoprime divisible by a repeated prime factor p^a with $a \geq 2$.

Lehmer [5] has shown that the only primes $p < 6.10^9$ satisfying this condition are $p = 1093$ and $p = 3511$, each in case with $w(p) = 2$. Since we require $p^2 < 10^{13}$, we restrict our attention to these two values of p . (It is easy to check directly that these are the only two such p up to $10^{6.5}$.) We have $f(1093^2) = 364$ and $f(3511^2) = 1755$. For each value of p we consider numbers $Q \equiv 1 \pmod{f(p^2)}$ such that $N = p^2 Q \leq X$ and for each such N test directly whether $2^N \equiv 2 \pmod{N}$.

We took $X = 10^{13}$. There are 23 pseudoprimes up to 10^{12} with a repeated factor and 54 up to 10^{13} : those up to 10^{12} are given in Table 4.

N	factors
1194649	1093^2
12327121	3511^2
3914864773	$29 \cdot 113 \cdot 1093^2$
5654273717	$1093^2 \cdot 4733$
6523978189	$43 \cdot 127 \cdot 1093^2$
22178658685	$5 \cdot 47 \cdot 79 \cdot 1093^2$
26092328809	$1093^2 \cdot 21841$
31310555641	$1093^2 \cdot 26209$
41747009305	$5 \cdot 29 \cdot 241 \cdot 1093^2$
53053167441	$3 \cdot 113 \cdot 131 \cdot 1093^2$
58706246509	$157 \cdot 313 \cdot 1093^2$
74795779241	$137 \cdot 457 \cdot 1093^2$
85667085141	$3 \cdot 11 \cdot 41 \cdot 53 \cdot 1093^2$
129816911251	$3511^2 \cdot 10531$
237865367741	$1093^2 \cdot 199109$
259621495381	$3511^2 \cdot 21061$
333967711897	$1093^2 \cdot 279553$
346157884801	$3511^2 \cdot 28081$
467032496113	$313 \cdot 1093^2 \cdot 1249$
575310702877	$337 \cdot 1093^2 \cdot 1429$
601401837037	$1093^2 \cdot 503413$
605767053061	$157 \cdot 313 \cdot 3511^2$
962329192917	$3 \cdot 29 \cdot 47 \cdot 197 \cdot 1093^2$

Table 4. The 23 pseudoprimes with repeated factor up to 10^{12} .

7 Checking Ranges by Sieving

We used a sieving technique to verify that the lists of pseudoprimes produced by the method of the preceding sections were complete in certain ranges.

Suppose that we wish to list those pseudoprimes in a range up to X which are divisible only by primes in some list \mathcal{L} of primes, all less than Y . Clearly we may assume that $Y \leq X$. We form a table indexed by the integers up to X and initially set each entry in the table to zero. For each p in \mathcal{L} we add $\log p$ into the table entries corresponding to numbers t with $t \equiv 0 \pmod{p}$ and $t \equiv 1 \pmod{f(p)}$: that is, $t \equiv p \pmod{pf(p)}$. At the end of this process we output any N for which the corresponding table entry is equal to $\log N$. Such an N has the property that all the prime factors p of N are in \mathcal{L} and that $N \equiv 1 \pmod{f(p)}$ for every p dividing N : that is, N is a pseudoprime whose prime factors are all in \mathcal{L} .

From Proposition 1(2) we note that taking $\mathcal{L} = \mathcal{B}[X]$ in the sieve will give all the pseudoprimes up to X .

To estimate the time taken to sieve over a range we need the following result.

Proposition 3. *Fix an integer b and let $f(p)$ denote the order of b in the multiplicative group modulo p for b prime to p . The sum, taken over primes p not*

dividing b ,

$$\sum_p \frac{1}{pf(p)}$$

is convergent.

Proof. Since the terms in the series are positive, the sum is convergent if any re-arrangement of it is convergent. Write

$$\sum_p \frac{1}{pf(p)} = \sum_f \frac{1}{f} \sum_{\substack{f(p)=f \\ p>b}} \frac{1}{p}.$$

If b has order f modulo p , then p divides $b^f - 1$ and $p \equiv 1 \pmod{f}$. Let p_i denote the k , say, distinct prime factors of $b^f - 1$ which satisfy $p_i > b$ and $p_i \equiv 1 \pmod{f}$: the p_i will include all the primes $p > b$ with $f(p) = f$. We have $k < f$ since all $p_i > b$, and $p_i \geq 1 + if$. So

$$\sum_{\substack{f(p)=f \\ p>b}} \frac{1}{p} \leq \sum_{i=1}^k \frac{1}{p_i} < \sum_{i=1}^f \frac{1}{if} < \frac{1}{f}(1 + \log f).$$

Hence

$$\sum_{p>b} \frac{1}{pf(p)} < \sum_f \frac{1 + \log f}{f^2}$$

and the latter sum is convergent. \square

We shall use this result with $b = 2$. In this case the numerical value of the sum, computed over the primes p up to 10^7 , is approximately 0.31734.

The time taken to sieve over all the numbers up to X will be bounded by

$$X + \sum_{p \in \mathcal{L}} \left\lceil \frac{X}{pf(p)} \right\rceil \leq X + X \sum_p \frac{1}{pf(p)} + \pi(Y) = O(X),$$

which is an improvement over a direct search for pseudoprimes: testing the condition $2^{N-1} \equiv 1 \pmod{N}$ for all N up to X would already take time $O(X \log X)$.

In practice, we found that the contribution of order $\pi(Y)$ from considering elements p of \mathcal{L} for which there are few or no multiples of $pf(p)$ in the range outweighs the contribution of order X from scanning the table and so it is beneficial to reduce the size of the list \mathcal{L} as much as possible.

We therefore consider a “large prime variation”. After sieving with \mathcal{L} the list of primes up to some limit Y , we use a further technique to deal with those pseudoprimes which have a prime factor q greater than Y . For each prime $q > Y$ in $\mathcal{B}[X]$, we consider all numbers P up to X/q which are $\equiv 1 \pmod{f(q)}$. The procedure now follows that of section 5. For each such P we test whether $2^{Pq} \equiv 2 \pmod{P}$. If so, $N = Pq$ is a pseudoprime.

8 Comparison with Existing Tables

We have checked our tables against those of Pomerance, Selfridge and Wagstaff [11], who obtained the 21853 pseudoprimes up to $25 \cdot 10^9$. We extracted the 19279 Carmichael numbers from our tables and compared them against the tables of [7]. In each case there was no discrepancy.

9 Some Details of the Computations

We ran the search procedure of the main strategy, sections 2 to 4, with upper limits of $X = 10^n$ for each value of n up to 13 and each value of d up to 9 independently. As a consequence the list of pseudoprimes up to 10^{12} was in effect computed twice, that up to 10^{11} three times and so on, providing additional checks on the computations. The computer programs were written in C and run on Sun 3/60 and Sparc workstations. The restriction of the search to prime factors less than $X/10^4$, that is, less than 10^9 , meant that 32-bit integer arithmetic could be used throughout. As a check, both on the programs and the results, some of the runs were duplicated using the rather strict Norcroft C compiler on an IBM 3084 mainframe. A total of about 2000 hours of CPU time was required. All the results were consistent.

The methods of sections 5 and 6 were implemented using Pari/GP on a Sparc workstation. Less than an hour was required for this part of the computation.

We used the sieving process of section 7 to check the search process up to 10^{12} : this consumed about 300 hours of CPU time on an IBM 3084. The results were consistent with those obtained by the methods of sections 2 and 3.

As a further check, we ran the “large prime variation” of §5 for pseudoprimes up to 10^{13} with a prime factor q in $\mathcal{B}[10^{13}]$ with $q > 10^7$: there are 39463 such primes. The lists matched those found by the search process: there were 3145 such pseudoprimes up to 10^{13} .

10 Statistics

Let $P(X)$ denote the number of pseudoprimes less than X , and $P(d, X)$ denote the number which have exactly d prime factors. In Table 1 we give $P(d, X)$ and $P(X)$ for values of X up to 10^{13} . No pseudoprime in this range has more than 9 prime factors. We have $P(10^{13}) = 264239$.

In Table 2 we give the smallest pseudoprime with d prime factors for d up to 17.

In Table 7 we give the number of pseudoprimes in each class modulo m for m up to 12.

In Tables 8 and 8 we give the number of pseudoprimes divisible by primes p up to 97. In Table 8 we count all pseudoprimes divisible by p : in Table 8 we count only those for which p is the smallest prime factor.

The largest prime factor of a pseudoprime up to 10^{13} is 77158673929, dividing

$$9799151588983 = 127 \cdot 77158673929$$

and the largest prime to occur as the smallest prime factor of a pseudoprime in this range is 3029563, dividing

$$9518187116947 = 3029563 \cdot 3141769.$$

Define $\alpha(X)$ by $P(X) = \exp(\log(X)^\alpha)$ and $\beta(X)$ by $P(X) = X\ell(X)^{-\beta}$, where

$$\ell(X) = \exp\left(\frac{\log X \log \log \log X}{\log \log X}\right).$$

Pomerance [8],[9],[10] showed that $\alpha \geq 85/207 > 0.4106$ and $\beta \geq \frac{1}{2}$ for X sufficiently large: he conjectured that β tends to 1. Clearly if β is even bounded then α tends to 1.

In Table 5 we tabulate the values of α and β for various values of X up to 10^{13} . We see that α is increasing over the range, but β is not obviously converging.

X	$\alpha(X)$	$\beta(X)$
10^3	0.048663	2.466690
10^4	0.508262	1.849388
10^5	0.602306	1.700002
10^6	0.649319	1.636881
10^7	0.679908	1.602218
10^8	0.697435	1.596159
10^9	0.711006	1.595093
10^{10}	0.721350	1.598918
$25 \cdot 10^9$	0.724828	1.601292
10^{11}	0.729621	1.605264
10^{12}	0.736643	1.612232
10^{13}	0.742721	1.619440

Table 5. The functions α and β of section 10.

Define an odd composite integer N to be an *Euler pseudoprime* if

$$2^{(N-1)/2} \equiv \left(\frac{2}{N}\right)$$

where $\left(\frac{2}{N}\right)$ is the Jacobi symbol. Further define N to be a *strong*, or *Miller–Rabin pseudoprime* if it passes the following test. Put $N - 1 = 2^a b$ with b odd, and form the sequence $2^b, 2^{2b}, \dots, 2^{2^a b} = 2^{N-1}$ modulo N . The test is passed if either the first term is 1 mod N or there are two consecutive terms -1 mod N , 1 mod N in the sequence. Finally define a *Carmichael number* to be a composite N for which $a^{N-1} \equiv 1$ mod N for any a prime to N .

It is clear that if N is an Euler pseudoprime or a strong pseudoprime then it is also a pseudoprime in the sense we have been using. (It is also true, but not quite so obvious, that if N is a strong pseudoprime then it is an Euler pseudoprime.) Since every Carmichael number is odd, it is again also a pseudoprime.

We can therefore tabulate the Euler pseudoprimes and strong pseudoprimes by extracting them from the tables of pseudoprimes. The Carmichael numbers in this range have already been tabulated in [7]. In Table 6 we give the numbers $EP(X)$, $SP(X)$ and $C(X)$ of Euler pseudoprimes, strong pseudoprimes and Carmichael numbers up to X for various values of X up to 10^{13} .

X	$P(X)$	$EP(X)$	$SP(X)$	$C(X)$
10^4	22	12	5	7
10^5	78	36	16	16
10^6	245	114	46	43
10^7	750	375	162	105
10^8	2057	1071	488	255
10^9	5597	2939	1282	646
10^{10}	14884	7706	3291	1547
$25 \cdot 10^9$	21853	11347	4842	2163
10^{11}	38975	20417	8607	3605
10^{12}	101629	53332	22412	8241
10^{13}	264239	124882	58897	19279

Table 6. The numbers of pseudoprimes, Euler pseudoprimes, strong pseudoprimes and Carmichael numbers up to X .

11 Even Pseudoprimes

The condition $2^{N-1} \equiv 1 \pmod{N}$ implies that N is odd. If we replace this condition by the closely related $2^N \equiv 2 \pmod{N}$ then it is possible for N to be even: for example, $N = 161038 = 2 \cdot 73 \cdot 1103$. Let us call such a number an *even pseudoprime*. It is easy to see that such an N satisfies $N = 2R$ with R odd and the condition becomes $2^{2R-1} \equiv 1 \pmod{R}$. It is then necessary that $f(R) \mid 2R - 1$, so $f(R)$ must be odd. Of the 145270 primes in $\mathcal{B}[10^{13}]$, 51607 have an odd value of f and so are candidates for being an odd prime factor of an even pseudoprime.

We adapted the methods of the previous sections to use this restricted set of possible prime factors and modified condition on N . There are only 155 even pseudoprimes up to 10^{12} : the 40 less than 10^{10} are listed in Table 9. We did not pursue this computation further.

Acknowledgements

The author is grateful to Prof. S.S. Wagstaff jr for providing files containing the tables described in [2] and [11]; to Prof. R. Heath-Brown and Dr W. Galway for discussions on Proposition 3; and to Prof. H. te Riele for discussions on even pseudoprimes.

m	c	$25 \cdot 10^9$	10^{11}	10^{12}	10^{13}
5	0	1474	2485	5695	13107
	1	12721	22936	61119	161588
	2	2743	4824	12643	32562
	3	2685	4768	12198	31381
	4	2230	3962	9974	25601
7	0	2025	3476	8546	20613
	1	8730	15868	42605	113703
	2	2049	3605	9407	24134
	3	2491	4387	11111	28742
	4	2039	3567	9178	23232
	5	2258	4030	10315	26717
	6	2261	4042	10467	27098
8	1	12654	22911	60415	158746
	3	1295	2180	5646	14522
	5	6615	11645	29902	76587
	7	1289	2239	5666	14384
9	1	11395	20644	54852	144736
	2	935	1649	4287	11107
	3	318	526	1117	2315
	4	3513	6148	15833	40994
	5	937	1634	4197	11025
	6	310	516	1134	2348
	7	3505	6209	15987	40745
	8	940	1649	4222	10969
11	0	1690	2930	7610	19271
	1	5314	9763	26416	70660
	2	1572	2773	7186	18399
	3	1554	2740	7090	18359
	4	1603	2739	7084	18273
	5	1776	3125	7806	20184
	6	1593	2886	7530	19482
	7	1709	3004	7667	19593
	8	1774	3114	8049	20740
	9	1428	2600	6727	17304
12	10	1840	3301	8464	21974
	1	16281	29360	77269	202532
	3	29	48	90	172
	5	2389	4202	10887	28310
	7	2132	3641	9403	23943
	9	599	994	2161	4491
11	11	423	730	1819	4791

Table 7. The number of pseudoprimes congruent to c modulo m .

p	$25 \cdot 10^9$	10^{11}	10^{12}	10^{13}	p	$25 \cdot 10^9$	10^{11}	10^{12}	10^{13}
3	628	1042	2251	4663	3	628	1042	2251	4663
5	1474	2485	5695	13107	5	1340	2278	5278	12315
7	2025	3476	8546	20613	7	1763	3044	7586	18452
11	1690	2930	7610	19271	11	1260	2203	5850	15192
13	2270	3997	9974	24836	13	1149	2147	5624	14486
17	1756	3018	7708	19572	17	654	1152	3100	8557
19	1530	2725	7129	18723	19	619	1099	2929	7777
23	671	1189	3137	8223	23	272	475	1277	3408
29	954	1717	4492	11943	29	345	628	1638	4414
31	1575	2783	7138	18322	31	551	966	2406	6035
37	1267	2286	5972	15542	37	301	531	1354	3613
41	1269	2238	5931	15579	41	237	444	1224	3288
43	930	1641	4296	11333	43	257	446	1081	2750
47	254	429	1091	2873	47	61	94	235	566
53	400	707	1878	4797	53	102	181	434	1096
59	145	246	631	1704	59	46	75	156	393
61	1007	1824	4897	13094	61	162	282	770	2119
67	486	830	2156	5793	67	103	171	433	1054
71	501	907	2502	6838	71	119	191	506	1226
73	1104	1990	5069	13296	73	135	246	614	1628
79	307	558	1432	3827	79	76	131	304	719
83	82	143	355	867	83	34	50	94	190
89	434	783	2098	5501	89	68	130	282	669
97	653	1147	2988	7779	97	105	179	389	911

Table 8. The number of times a prime $p \leq 97$ occurs in a pseudoprime, as any factor and as the least prime factor respectively.

Some of the work for this paper was carried out while the author held a Max Newman research fellowship in the Department of Pure Mathematics and Mathematical Statistics, University of Cambridge. The author is grateful to the Department and to the Cambridge University Computer Laboratory for the use of their computing facilities.

N	factors
161038	$2 \cdot 73 \cdot 1103$
215326	$2 \cdot 23 \cdot 31 \cdot 151$
2568226	$2 \cdot 23 \cdot 31 \cdot 1801$
3020626	$2 \cdot 7 \cdot 359 \cdot 601$
7866046	$2 \cdot 23 \cdot 271 \cdot 631$
9115426	$2 \cdot 31 \cdot 233 \cdot 631$
49699666	$2 \cdot 311 \cdot 79903$
143742226	$2 \cdot 23 \cdot 31 \cdot 100801$
161292286	$2 \cdot 127 \cdot 199 \cdot 3191$
196116194	$2 \cdot 127 \cdot 599 \cdot 1289$
209665666	$2 \cdot 7 \cdot 89 \cdot 191 \cdot 881$
213388066	$2 \cdot 23 \cdot 31 \cdot 151 \cdot 991$
293974066	$2 \cdot 73 \cdot 631 \cdot 3191$
336408382	$2 \cdot 73 \cdot 1103 \cdot 2089$
377994926	$2 \cdot 23 \cdot 89 \cdot 127 \cdot 727$
410857426	$2 \cdot 7 \cdot 191 \cdot 153649$
665387746	$2 \cdot 23 \cdot 3463 \cdot 4177$
667363522	$2 \cdot 7 \cdot 5471 \cdot 8713$
672655726	$2 \cdot 73 \cdot 1103 \cdot 4177$
760569694	$2 \cdot 1319 \cdot 288313$
1066079026	$2 \cdot 23 \cdot 31 \cdot 151 \cdot 4951$
1105826338	$2 \cdot 23 \cdot 73 \cdot 127 \cdot 2593$
1423998226	$2 \cdot 7 \cdot 79 \cdot 271 \cdot 4751$
1451887438	$2 \cdot 79 \cdot 89 \cdot 223 \cdot 463$
1610063326	$2 \cdot 73 \cdot 2089 \cdot 5279$
2001038066	$2 \cdot 47 \cdot 311 \cdot 68449$
2138882626	$2 \cdot 73 \cdot 3191 \cdot 4591$
2952654706	$2 \cdot 31 \cdot 71 \cdot 631 \cdot 1063$
3220041826	$2 \cdot 73 \cdot 103 \cdot 233 \cdot 919$
3434672242	$2 \cdot 727 \cdot 911 \cdot 2593$
4338249646	$2 \cdot 4721 \cdot 459463$
4783964626	$2 \cdot 7 \cdot 23 \cdot 73 \cdot 271 \cdot 751$
5269424734	$2 \cdot 7 \cdot 1433 \cdot 262657$
5820708466	$2 \cdot 79 \cdot 3257 \cdot 11311$
6182224786	$2 \cdot 23 \cdot 31 \cdot 151 \cdot 28711$
6381449614	$2 \cdot 73 \cdot 199 \cdot 239 \cdot 919$
8356926046	$2 \cdot 7 \cdot 79 \cdot 7555991$
8419609486	$2 \cdot 31 \cdot 2441 \cdot 55633$
9548385826	$2 \cdot 7 \cdot 31 \cdot 89 \cdot 247201$
9895191538	$2 \cdot 127 \cdot 1289 \cdot 30223$

Table 9. The 40 even pseudoprimes up to 10^{10} .

References

- [1] D.M. Bressoud, *Factorization and primality testing*, Springer–Verlag, New York, 1989.
- [2] J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman, and S.S. Wagstaff jr, *Factorizations of $b^n \pm 1$* , second ed., Contemporary mathematics, vol. 22, Amer. Math. Soc., Providence RI, 1988.
- [3] M. Cipolla, *Sui numeri composti p , che verificano la congruenza di Fermat $a^{P-1} \equiv 1 \pmod{p}$* , Annali di Mathematica Pura e Applicata **9** (1904), 139–160.
- [4] N. Koblitz, *A course in number theory and cryptography*, Graduate Texts in Mathematics, vol. 114, Springer–Verlag, New York, 1987.
- [5] D.H. Lehmer, *On Fermat's quotient, base two*, Math. Comp. **36** (1981), 289–290.
- [6] R.A. Mollin (ed.), *Number theory and its applications*, Dordrecht, Kluwer Academic, 1989, Proceedings of the NATO Advanced Study Institute on Number Theory and Applications.
- [7] R.G.E. Pinch, *The Carmichael numbers up to 10^{15}* , Math. Comp. **61** (1993), 381–391, Lehmer memorial issue.
- [8] C. Pomerance, *On the distribution of pseudoprimes*, Math. Comp. **37** (1981), 587–593.
- [9] ———, *A new lower bound for the pseudoprime counting function*, Illinois J. Maths **26** (1982), 4–9.
- [10] ———, *Two methods in elementary analytic number theory*, In Mollin [6], Proceedings of the NATO Advanced Study Institute on Number Theory and Applications.
- [11] C. Pomerance, J.L. Selfridge, and S.S. Wagstaff jr, *The pseudoprimes up to $25 \cdot 10^9$* , Math. Comp. **35** (1980), no. 151, 1003–1026.
- [12] P. Ribenboim, *The little book of big primes*, Springer–Verlag, New York, 1991.
- [13] ———, *The new book of prime number records*, third ed., Springer–Verlag, New York, 1996.
- [14] H. Riesel, *Prime numbers and computer methods for factorization*, second ed., Progress in mathematics, vol. 126, Birkhauser, Boston, 1994.

Computing the Number of Goldbach Partitions up to $5 \cdot 10^8$

Jörg Richstein

Institut für Informatik
Justus-Liebig-Universität
Gießen, Germany

Joerg.Richstein@informatik.uni-giessen.de

Abstract. Computing the number of Goldbach partitions

$$g(n) = \#\{(p, q) \mid n = p + q, p \leq q\}$$

of all even numbers n up to a given limit can be done by a very simple, but space-demanding sequential procedure. This work describes a distributed implementation for computing the number of partitions with minimal space requirements. The program was distributed to numerous workstations, leading to the calculation of $g(n)$ for all even n up to 5×10^8 . The resulting values are compared to those following from previously stated conjectures about the asymptotic behaviour of g .

1 Introduction

One of the most famous unsolved problems in number theory, the *Goldbach Conjecture* states that every even number can be written as the sum of two primes. While still being unproved that every even number n has at least one partition (p, q) with $n = p + q$, it has long been observed that the number of partitions grows with increasing n . Table 1 shows a few values of g .

n	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50
$g(n)$	1	1	1	2	1	2	2	2	2	3	3	3	2	3	2	4	4	2	3	4	3	4	5	4

Table 1.

The value $g(n)$ strongly depends on the factorization of n . As an example, $120 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 5$ yields $g(120) = 12$, whereas the neighbouring even numbers $118 = 2 \cdot 59$ and $122 = 2 \cdot 61$ give $g(118) = 6$ and $g(122) = 4$, respectively.

Heuristical explanations, and formulas for $g(n)$ based on probabilistic considerations have been derived by numerous authors in the past; a nice representation, given by Nils Pipping in 1926 can be found in [24]: By sieving out the primes from the sets $\{3, 5, 7, 9, \dots, n\}$ and $\{n - 3, n - 5, n - 7, n - 9, \dots, 3\}$, one

can get a first approximation to $g(n)$ by taking $g(n) \approx n \cdot P^2(n)$, where $P(n)$ denotes the probability that a number less than n is prime. But twice choosing a prime is not independent from each other, so a correction will be necessary: By first considering those p that divide n and then those being coprime to n , one can get two correction factors,

$$\prod_{\substack{3 \leq p \leq \sqrt{n} \\ p \nmid n}} \frac{p}{p-1}$$

for the first case and

$$\prod_{\substack{p \leq \sqrt{n} \\ p \nmid n}} \frac{p(p-2)}{(p-1)^2}$$

for the second. Multiplying both gives

$$2 \prod_{\substack{3 \leq p \leq \sqrt{n} \\ p \nmid n}} \frac{p-1}{p-2} \cdot \prod_{3 \leq p \leq \sqrt{n}} \frac{p(p-2)}{(p-1)^2},$$

where the second product tends to the *twin prime constant* $C_2 \approx 0.66016182$. Thus, it can be conjectured that

$$g(n) \sim 2C_2 P^2(n) \prod_{\substack{3 \leq p \leq \sqrt{n} \\ p \nmid n}} \frac{p-1}{p-2}, \quad (1)$$

where the quotients of the product explain $g(n)$'s dependency on n 's factors. For details on the derivation of the above two factors, the reader is referred to Pipping's description in [24].

In 1871, Sylvester [43] was the first one who described a formula close to (1). Since then, many authors have suggested different formulas based on (1) with different substitutions of the function $P(n)$ and sometimes different correction factors. In 1974, Halberstam and Richert [14] proved that

$$g(n) \leq 4C_2 \prod_{\substack{3 \leq p \leq \sqrt{n} \\ p \nmid n}} \frac{p-1}{p-2} \frac{n}{\log^2 n} \cdot \left(1 + O\left(\frac{\log \log n}{\log n}\right) \right).$$

More recently, in 1993, Deshouilliers, Granville, Narkiewicz and Pomerance showed that the maximal n for which equality holds in

$$g(n) \leq \pi(n-2) - \pi(n/2-1)$$

is 210.

In Section 4.1, most of the formulas trying to give an exact estimation to g will be revisited and a statistical comparison to the computed values of g will be made. Section 2.1 gives an overview of past computations. In Section 3, our distributed implementation will be described and practical considerations and running times will be given. Finally, a discussion on the results and method follows.

2 Computing Values of g

2.1 Haussner's "Strip Machine"

In 1896, Robert Haussner described a mechanical way to obtain values of g . We will shortly give a translation of Haussner's original description [18] of his "partition counting machine":

"May I be permitted to briefly demonstrate how to construct an apparatus by which one can obtain all partitions of an even number $\leq 2N$ without any calculation. One writes all odd numbers from 1 through $2N - 1$ equidistantly on two parallel strips, on one strip in ascending, on the other in descending order. The prime numbers are somehow emphasized on both strips. If one moves both strips lengthwise such that the number 1 of the first strip lies opposite the number $2n - 1$ of the second strip, where $n \leq N$, then all cases in which two prime numbers face each other give all partitions of $2n$ in two prime number summand; yet one only has to consider those prime numbers on the first strip that are $\leq n$. For greater convenience, after adjustment both strips should be reeled off one roll and wound onto another. It is easy to attach a mechanical counter that displays the number ν^1 once unwinding has been successfully completed."

Algorithmically, Haussner's strip machine can be summarized as follows:

Algorithm 1 Haussner's strip machine

Input: Upper limit $2N$

- 1: $oddpbit \leftarrow sieve(oddpbit, 2N)$
- 2: $revpbit \leftarrow reverse(oddpbit)$
- 3: **for** $n \leftarrow N$ **downto** 3 **do**
- 4: $g2n \leftarrow 0$
- 5: **for** $i \leftarrow 1$ **to** $\lceil \frac{n}{2} \rceil$ **do**
- 6: **if** $oddpbit[i] = 1$ **and** $revpbit[i] = 1$ **then**
- 7: $g2n \leftarrow g2n + 1$
- 8: **end if**
- 9: **end for**
- 10: $output(g2n)$
- 11: $revpbit \leftarrow shiftleft(revpbit)$
- 12: **end for**

Here, the strips have been substituted by two bit-arrays $oddpbit$ and $revpbit$. After being sieved by the function *sieve*, $oddpbit$ represents the odd numbers up to $2N$ such that $oddpbit[i] = 1$ iff $2i - 1$ is prime. Its reversed counterpart $revpbit$ is equal to 1 at position i iff $2(n - i) + 1$ is prime. The readjustment of the second

¹ $= g(2n)$

strip of Haussner's machine is realized by the function *shiftleft*, which shifts the whole array *revbit* left by one bit.

Algorithm 1 could easily be implemented as a computer program. A few practical notes should be added, though. Instead of using two strips, it would be sufficient to only use the array *oddpbit*, successively checking (in line 6) if *oddpbit*[i] and *oddpbit*[$N - i + 1$] are simultaneously equal to 1. But the use of two bit-arrays can be advantageous, because one would in practice pack the bits representing the odd numbers into computer-words. After joining the relevant elements of the two arrays wordwise by binary AND, one only needs to count the remaining 1-bits in order to get the value of $g(2n)$.

The generation of the primes below $2N$ will require at most $O(N \log \log N)$ operations and N bits of space. For each $3 \leq n \leq N$, the inner loop is executed $\lceil n/2 \rceil$ times, so Algorithm 1 computes the number of all Goldbach partitions of all even numbers up to $2N$ in $O(N^2)$ operations and $O(N)$ bits of space. It should be mentioned that in practice the bit-shifting "hidden" in the function *shiftleft* does affect the running time if one packs the odd numbers into words, because three operations are necessary to shift one word.

Although Haussner published extensive tables including the number of Goldbach partitions of all even numbers up to 5000 in 1896 [19], he never built his machine. Instead, he calculated his tables in a way similar² to the following one, which is basically the method from which our distribution will be derived.

2.2 The Base Method

Algorithm 2 Base method

Input: Upper limit $2N$

```

1: oddp  $\leftarrow$  genprimes( $2N$ )
2: for  $i \leftarrow 1$  to  $\pi(N) - 1$  do
3:    $j \leftarrow i$ 
4:   while oddp[ $i$ ] + oddp[ $j$ ]  $\leq 2N$  do
5:      $g[\text{oddp}[i] + \text{oddp}[j]] \leftarrow g[\text{oddp}[i] + \text{oddp}[j]] + 1$ 
6:      $j \leftarrow j + 1$ 
7:   end while
8: end for
```

Output: g

In Algorithm 2, *oddp* is an array containing all odd primes up to $2N$. While the space requirements are still $O(N)$, Algorithm 2 only needs $O(N^2 / \log^2 N)$ steps to generate the array g . However, storing the primes up to $2N$ directly in the array *oddp* requires approximately twice as much memory than that needed for the array *oddpbit* in Algorithm 1. So it would be appropriate to replace

² Haussner actually used a method more suitable for hand calculation, based on residue tables mod 100 (see [19] for details).

oddp by another array, say *oddpdiff*, storing (half) differences between successive primes instead³. This way, the memory requirements of the arrays *oddpbit* and *oddpdiff* would be about the same in our range. But the most space consuming part is the array *g* itself. In the range considered here, already three bytes of memory are necessary to store a single value of *g*. Since one must keep all intermediate values of *g* for all $2n \leq 2N$ as well as the array *oddpdiff* in main memory during the whole computation, the needed memory sums up to about $4N + 2N/\log 2N$ Bytes (assuming that each value will be stored in a 32-bit integer). So, for example, a computation up to $5 \cdot 10^8$ would require approximately 1 Gigabyte of main memory. At the time this work was carried out, this amount was only available on machines that could not be occupied for long running computations.

Two years prior to Haussner, Cantor [9] had published a table with all partitions of all $2n$ below 1000, extended to 2000 by Aubry [1] in 1896. In an unpublished work in 1917, Weinreich [38], [45] checked Haussner's tables, making use of Haussner's idea of utilizing paper strips. In 1927, Pipping [25] published a corrected list of Haussner's results up to 5000, which he had produced with a modular method⁴ suitable for hand calculation (again involving paper strips, see [24]). Pipping's computations were further extended by Stein and Stein to 200000 in 1964 and by Bohman and Fröberg [3] to 350000 in 1975 (without explicitly describing their ways of computation).

Almost exactly 100 years after Haussner's description, Lavenier and Saouter [22] were the first ones to construct an impressive "version" of his partition counting machine, using a dedicated hardware being capable of computing 100 values of *g* simultaneously. Their computations went up to $1.28 \cdot 10^8$ in 1998.

3 Distributing the Base Algorithm

In addition to its inferior running time, there is no apparent way to distribute Algorithm 1, so one would rather think about finding a distributed version of Algorithm 2.

The major problem of Algorithm 2 is its huge memory expense, which can be lowered by a time/space-trade that will now be described.

3.1 Principle

In preparation to distributing Algorithm 2, let $2N = 2^t \cdot \Pi_m$, where Π_m is the product of the first m odd primes. The interval $[1, 2N]$ will now be divided into 2^{t-r} segments $s_0, s_1, \dots, s_{2^{t-r}-1}$ where $t > r \geq 5$ so each segment has a length of $2^r \cdot \Pi_m$. Then, the primes p, q of each partition (p, q) of an even number in the segment s_i ($0 \leq i \leq 2^{t-r}$) will origin in segments as shown in Table 2. The principle of the distributed version of Algorithm 2 is now as follows.

³ This requires only one byte per prime number up to approximately $3 \cdot 10^{11}$.

⁴ Both Haussner's and Pipping's calculation methods don't give an advantage if computers are available.

$p \in$	$q \in$
s_0	$s_{i-1} \cup s_i$
s_1	$s_{i-2} \cup s_{i-1}$
s_2	$s_{i-3} \cup s_{i-2}$
\dots	\dots
s_j	$s_{i-j-1} \cup s_{i-j}$
\dots	\dots
$s_{\lfloor i/2 \rfloor}$	$s_{i-\lfloor i/2 \rfloor-1} \cup s_{i-\lfloor i/2 \rfloor}$

Table 2. Possible subsets of partitions

For each segment $s_i \subset [1, 2N]$, s_j and s_{i-j-1} as well as s_{i-j} will be sieved for primes for all $j \in [0, \lfloor i/2 \rfloor]$. For every j , all sums $p + q$, $p \in s_j$, $q \in s_{i-j-1} \cup s_{i-j}$ will be formed and checked for being $\in s_i$. If so, g will be incremented by 1 at position $(p+q-i \cdot 2^r \Pi_m)/2$. After a segment s_i has been completely processed, an array element $g[n]$ will hold the number of partitions of the number $i \cdot 2^r \Pi_m + 2n$. During this process, two things have to be kept in mind: Firstly, no additions must be performed when $j > i - j - 1$, and secondly, if $j = i - j - 1$ or $j = i - j$, one must additionally check that $p \leq q$. Also note that the sieving of two higher segments is only necessary when $j = 0$. In all other cases, the second highest segment will become the highest in the next step and so on.

3.2 Implementation

Algorithm 3 Processing one segment

Input: Segment number i , $i > 1$

- 1: $(oddph_i, \pi_{hi}) \leftarrow genprimes(s_i)$ { Get primes in highest segment }
 - 2: **for** $j \leftarrow 0$ **to** $\lfloor i/2 \rfloor - 1$ **do** { Process segments 0 through $\lfloor i/2 \rfloor - 1$ }
 - 3: $(oddpl_o, \pi_{lo}) \leftarrow genprimes(s_j)$ { Get primes in lower segment }
 - 4: $g \leftarrow add_{hi}(g, i, oddpl_o, \pi_{lo}, oddph_i, \pi_{hi})$ { Add lower/higher segment }
 - 5: $(oddph_i, \pi_{hi}) \leftarrow genprimes(s_{i-j-1})$ { Get primes in 2^{nd} higher segment }
 - 6: $g \leftarrow add_{lo}(g, i, oddpl_o, \pi_{lo}, oddph_i, \pi_{hi})$ { Add lower/ 2^{nd} higher segment }
 - 7: **end for**
 - 8: **if** i is odd **then**
 - 9: $(oddpl_o, \pi_{lo}) \leftarrow genprimes(s_{j+1})$ { Get primes in middle segment }
 - 10: $g \leftarrow add_{hi}(g, i, oddpl_o, \pi_{lo}, oddph_i, \pi_{hi})$ { Add lower/higher segment }
 - 11: $g \leftarrow add_{mid}(g, i, oddpl_o, \pi_{lo})$ { Add middle segment }
 - 12: **else** { i is even }
 - 13: $g \leftarrow add_{mid}(g, i, oddph_i, \pi_{hi})$ { Add middle segment }
 - 14: **end if**
- Output:** g
-

The function *genprimes* generates the primes of a segment, returning them along with $\pi_{lo/hi}$, the number of primes found in the segment. In order to optimize the additions of two segments, three different functions *add_{lo}*, *add_{mid}*, *add_{hi}* are called, depending on which segments are to be processed:

Algorithm 4 add_{lo}

Input: Array g , segment number i , segments $oddpi_{lo}$, $oddph_{hi}$, π_{lo} , π_{hi}

- 1: **for** $j \leftarrow \pi_{lo}$ **downto** 1 **do** { Process primes p of $oddpi_{lo}$ }
- 2: $p \leftarrow oddpi_{lo}[j]$
- 3: **for** $k \leftarrow \pi_{hi}$ **downto** 1 **do** { Process primes q of $oddph_{hi}$ }
- 4: $q \leftarrow oddph_{hi}[k]$
- 5: **if** $p + q \notin s_i$ **then** { No more q for this p }
- 6: **break** { (for k ...) }
- 7: **end if**
- 8: $g[(p + q - i \cdot 2^r \Pi_m)/2] \leftarrow g[(p + q - i \cdot 2^r \Pi_m)/2] + 1$
- 9: **end for**
- 10: **end for**

Output: g

Algorithm 5 add_{hi}

Input: Array g , segment number i , segments $oddpi_{lo}$, $oddph_{hi}$, π_{lo} , π_{hi}

- 1: **for** $j \leftarrow 1$ **to** π_{lo} **do** { Process primes p of $oddpi_{lo}$ }
- 2: $p \leftarrow oddpi_{lo}[j]$
- 3: **for** $k \leftarrow 1$ **to** π_{hi} **do** { Process primes q of $oddph_{hi}$ }
- 4: $q \leftarrow oddph_{hi}[k]$
- 5: **if** $p + q \notin s_i$ **then** { No more q for this p }
- 6: **break** { (for k ...) }
- 7: **end if**
- 8: $g[(p + q - i \cdot 2^r \Pi_m)/2] \leftarrow g[(p + q - i \cdot 2^r \Pi_m)/2] + 1$
- 9: **end for**
- 10: **end for**

Output: g

Algorithm 6 add_{mid}

Input: Array g , segment number i , segment $oddpm_{mid}$, π_{mid}

- 1: **for** $j \leftarrow 1$ **to** π_{mid} **do** { Process primes p of $oddpm_{mid}$ }
- 2: $p \leftarrow oddpm_{mid}[j]$
- 3: **for** $k \leftarrow j$ **to** π_{mid} **do** { Process primes q of $oddpm_{mid}$ }
- 4: $q \leftarrow oddpm_{mid}[k]$
- 5: **if** $p + q \in s_i$ **then** { If so, increment g }
- 6: $g[(p + q - i \cdot 2^r \Pi_m)/2] \leftarrow g[(p + q - i \cdot 2^r \Pi_m)/2] + 1$
- 7: **end if**
- 8: **end for**
- 9: **end for**

Output: g

3.3 Practical Considerations and Running Times

The reason for choosing the length of the segments to be $2^r \Pi_m$ is that one can efficiently apply a segmented sieve as given in [5] or [2] in order to generate the primes in the segments. We choosed $r \geq 5$, so the segment lengths are divisible

by 32, which was the base word length. The actual implementation of the sieve is described in [30].

The running time of Algorithm 3 is essentially determined by the addition operations. Each call to one of the adding functions will cause at most $c \cdot l^2 / \log^2 l$ operations, where $l = 2^r \Pi_m$ abbreviates the segment length and c is a constant. So for one segment s_i this will sum up to $O(i \cdot l^2 / \log^2 l)$ operations. Therefore, the processing of all segments (starting with s_2) will take about

$$\sum_{i=2}^{\frac{2N}{l}} i \cdot \frac{l^2}{\log^2 l} = \sum_{i=2}^{\frac{2N}{l}} i \cdot \frac{(2N)^2}{\log^2 \frac{2N}{i}} = 4N^2 \sum_{i=2}^{\frac{2N}{l}} \frac{1}{i \cdot \log^2 \frac{2N}{i}} .$$

By approximating the sum with

$$\int_2^{\frac{2N}{l}} \frac{dx}{x \log^2 \frac{2N}{x}} = \left[\frac{1}{\log \frac{2N}{x}} \right]_2^{\frac{2N}{l}} = \frac{1}{\log l} - \frac{1}{\log N} ,$$

we finally get that the number of all partitions can be determined in $O(N^2 / \log l)$ operations. This suggests to take the segment length as large as possible (which was expectable).

The space requirements of Algorithm 3 are determined by the array g , which now takes $O(l)$ bits of space. An additional $O(l)$ bits is needed for the two prime arrays *oddp* and the space for the sieves, but this is in practice negligible.

In the actual implementation, we took $m = 5$, $r = 6$ and $2N = 500660160$, so $l = 2^6 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 960960$, giving a total space of around 2MB. The program was distributed to 7 Sun Ultra1, 6 Sun4 workstations and 3 PC's. The total running time was approximately 70 days. Due to the quadratic running time, processing the last segments already took about 3 days.

4 Results

The resulting values of g have been checked against those of [22] up to $1.28 \cdot 10^8$, without finding any discrepancies.

There is a relatively easy possibility to check the resulting values by using an “adapted version” of Landau’s [21] summatory function

$$H(x) = \sum_{n=1}^x G(n) ,$$

where $G(n) = \#\{(p, q) \mid n = p + q\}$ (so for even n one has $g(n) = \lceil G(n)/2 \rceil$). Landau also defined H for odd numbers, giving $G(2k+1) = 1$, if $2k+1$ is prime and 0 otherwise.

In our case, we define

$$h(x) = \sum_{\substack{6 \leq n \leq x \\ n \text{ even}}} g(n) .$$

Since

$$\begin{aligned}
h(x) &= \sum_{3 \leq p \leq \frac{x}{2}} \sum_{p \leq q \leq x-p} 1 \\
&= \sum_{3 \leq p \leq \frac{x}{2}} (\pi(x-p) - \pi(p) + 1) \\
&= \sum_{3 \leq p \leq \frac{x}{2}} \pi(x-p) - \sum_{3 \leq p \leq \frac{x}{2}} \pi(p) + \sum_{3 \leq p \leq \frac{x}{2}} 1 \\
&= \sum_{3 \leq p \leq \frac{x}{2}} \pi(x-p) - \frac{\pi(\frac{x}{2})(\pi(\frac{x}{2})+1)}{2} + 1 + \pi(\frac{x}{2}) - 1 \\
&= \sum_{3 \leq p \leq \frac{x}{2}} \pi(x-p) - \frac{\pi^2(\frac{x}{2}) - \pi(\frac{x}{2})}{2} ,
\end{aligned}$$

one can check the computation by calculating the sum on the last line, subtract the quotient and compare the result to the computed $h(x)$.⁵

For $x = 500660160$ it turned out that $\sum_{3 \leq p \leq \frac{x}{2}} \pi(x-p) = 277532324737949$, $(\pi^2(x/2) - \pi(x/2))/2 = 93795323525751$ and therefore $h(x) = 183737001212198$ which was identical to the directly computed $\sum_{6 \leq n \leq x} g(n)$. The checking program for the summation of the $\pi(x-p)$ was based on the same sieve program as used above, appropriately modified.

Figures 1 and 2 show plots of $g(n)$ for even n up to 160160 and between 500500000 and 500660160, where the stronger lines visible correspond to numbers divisible by smaller primes (as a consequence of the quotients $\frac{p-1}{p-2}$ of (1)).

The maximal value of $g(n)$ in the range considered was 3977551, taken at $n = 497668710 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23 \cdot 29$.

In [41], it was conjectured that $g : 2\mathbb{N} \rightarrow \mathbb{N}$ is surjective. Up to 500660160, the smallest n that didn't occur as a value of g was 2166940.

4.1 Comparisons to Previously Stated Conjectures on g

Sylvester in 1871 was the first one to formulate a conjecture about the asymptotic behaviour of g . Here, we will shortly revisit his formula along with the ones that have been suggested since then and later compare them to the computed values of g . In the following, g_X will denote a suggested formula meaning $g_X(n) \sim g(n)$, where the index X will abbreviate the author's name.

Only given in a short abstract in [43], Sylvester describes⁶

$$g_{Sy}(n) = \frac{n}{\log^2 n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p|n}} \frac{p-1}{p-2} .$$

⁵ Actually this is only a 1-error-detection.

⁶ It is not quite clear whether Sylvester exactly meant g_{Sy} . The interpretation of the abstract [43] is due to Shah/Wilson [35].

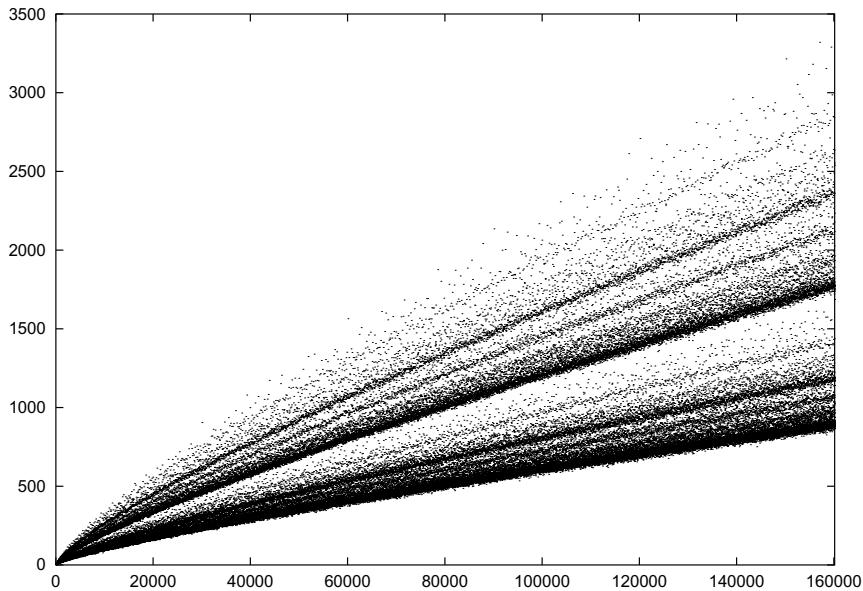


Fig. 1. $g(n), 6 \leq n \leq 160160$

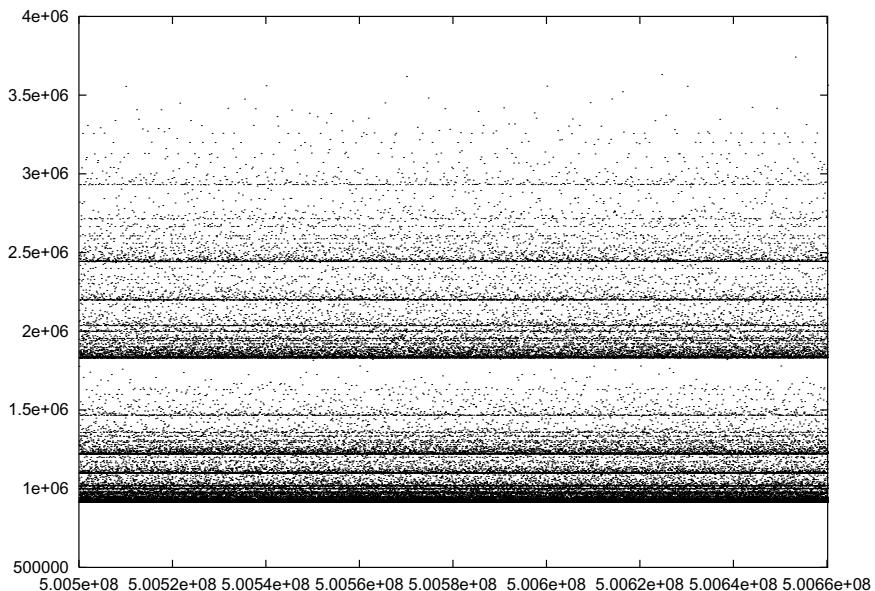


Fig. 2. $g(n), 500500000 \leq n \leq 500660160$

Without knowing of Sylvester's work, Stäckel suggested in 1896 [36] two different (asymptotically equal) formulas

$$g_{St1}(n) = \frac{\pi^2(n)}{\varphi(n)} ,$$

and

$$g_{St2}(n) = \frac{(\pi(n - \sqrt{n}) - \pi(\sqrt{n}))^2}{n - 2\sqrt{n}} \cdot \frac{n}{\varphi(n)} ,$$

where φ is Euler's function. Without saying that Stäckel's formulas are at all right, Landau [21] shows indirectly in 1900 that a correction factor $\pi^4/105\zeta(3)$ must in any case be multiplied, leading to

$$g_{La}(n) = 0.772 \dots \cdot \frac{\pi^2(n)}{\varphi(n)} .$$

In 1915, Brun [6] gave

$$g_{Br}(n) = 1.5985 \cdot \frac{n}{\log^2 n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p \mid n}} \frac{p-1}{p-2} .$$

One year later, Stäckel [37] corrects this to

$$g_{St}(n) = 2 \cdot \prod_{p>2} \left(1 - \frac{1}{(p-1)^2}\right) \cdot \frac{\pi^2(n)}{n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p \mid n}} \frac{p-1}{p-2} ,$$

where the first product is identical to the second factor C_2 of (1). Unfortunately, Stäckel is mostly cited for his attempt g_{St1} , although he was apparently the first one to give the most likely asymptotically correct formula g_{St} . Before 1919, Hardy and Littlewood (published in 1919 [15]), without knowing Stäckel's work of 1916⁷, gave a formula asymptotically equal to Stäckel's:

$$g_{HL}(n) = 2C_2 \frac{n}{\log^2 n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p \mid n}} \frac{p-1}{p-2} ,$$

After being requested by Hardy and Littlewood to check their formula, Shah and Wilson [35] replace $\log^2 n$ by $\log^2 n - 2 \log n$:

$$g_{SW1}(n) = 2C_2 \frac{n}{\log^2 n - 2 \log n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p \mid n}} \frac{p-1}{p-2} ,$$

⁷ Hardy and Littlewood mention in 1922 [17] that they "have until very recently been unable to consult" it.

after first suggesting

$$g_{SW2}(n) = 2C_2 \frac{n}{\log^2 n - 2 \log n + 2 - \frac{\pi^2}{6}} \prod_{\substack{2 < p \leq \sqrt{n} \\ p|n}} \frac{p-1}{p-2}$$

Finally in 1942, Selmer [33] gave an average approximation

$$g_{Se1}(n) = 2C_2 \int_0^{\frac{n}{2}} \frac{dx}{\log(\frac{n}{2}+x) \log(\frac{n}{2}-x)} \prod_{\substack{2 < p \leq \sqrt{n} \\ p|n}} \frac{p-1}{p-2},$$

which he found by taking the derivate of Landau's summatory function, replacing $n/\log n$ by $li(n)$. Selmer also considered the second term of Riemann's approximation to $\pi(x)$, leading to $g_{Se2}(n) =$

$$C_2 \int_0^{\frac{n}{2}} \frac{1}{\log(\frac{n}{2}+x) \log(\frac{n}{2}-x)} \cdot \left(2 - \left(\frac{1}{\sqrt{\frac{n}{2}+x}} + \frac{1}{\sqrt{\frac{n}{2}-x}} \right) \right) dx \prod_{\substack{2 < p \leq \sqrt{n} \\ p|n}} \frac{p-1}{p-2}.$$

In the following comparison, we will also consider exchanging $\pi^2(n)$ in Stäckel's formula by the square of Riemann's

$$R(x) = \sum_{k=1}^{\infty} \frac{\mu(k)}{k} Li(x^{\frac{1}{k}}) .$$

Values of R can be easily computed by taking Gram's formula

$$R(x) = 1 + \sum_{k=1}^{\infty} \frac{\log^k x}{k! k \zeta(k+1)} ,$$

and using precomputed values of $\zeta(k)$. In our range, we took 70 iterations of the sum. So we get "Riemann's approximation" to g :

$$g_{R(n)}(n) = 2C_2 \frac{R^2(n)}{n} \prod_{\substack{2 < p \leq \sqrt{n} \\ p|n}} \frac{p-1}{p-2} .$$

(which we will denote by $g_{R(n)}$ in order not to accidentally imply that Riemann himself ever gave it).

In Tables 3, 4 and 5, we list the real errors of the corresponding summatory functions $h_X(2N)$ along with their relative errors and the average, absolute, relative errors

$$\frac{1}{N-2} \cdot \sum_{6 \leq n \leq 2N} \frac{|g_X(n) - g(n)|}{g(n)} ,$$

where $2N$ denotes our upper limit 500660160. Due to their high computational

Table 3. ($h(2N) = 183737001212198$)

X	$h_X(2N) - h(2N)$	$\frac{h_X(2N) - h(2N)}{h(2N)}$	$\frac{1}{N-2} \sum \frac{ g_X(n) - g(n) }{g(n)}$
Sy	65622453672945	35.70	35.28
HL	-19119410676636	-10.41	10.69
$SW2$	-482589183065	-0.26	0.28
$R(n)$	375967802411	0.20	0.22
$SW1$	-291436375645	-0.16	0.17

Table 4. ($h(10^6) = 1671879782$)

X	$h_X(10^6) - h(10^6)$	$\frac{h_X(10^6) - h(10^6)}{h(10^6)}$	$\frac{1}{499998} \sum \frac{ g_X(n) - g(n) }{g(n)}$
Sy	473784719	28.34	27.51
HL	-255394009	-15.28	15.83
$SW2$	-7400835	-0.44	1.03
$R(n)$	7885603	0.47	1.00
St	7936767	0.47	0.99
$Se1$	5435703	0.33	0.99
$SW1$	-3419480	-0.20	0.93
$Se2$	352560	0.02	0.89

Table 5. ($\Delta h = 457957086522$)

X	$\Delta h - \Delta h_X$	$(\Delta h_X - \Delta h)/\Delta h$	$\frac{1}{330081} \sum \frac{ g_X(n) - g(n) }{g(n)}$
Sy	165533832724	36.15	36.15
HL	-46352189108	-10.12	10.12
$SW2$	-1145470091	-0.250	0.250
St	881922554	0.193	0.193
$R(n)$	869907910	0.190	0.190
$SW1$	-696379629	-0.152	0.151
$Se1$	29737472	0.006	0.041
$Se2$	-13166618	-0.003	0.040

costs, we only compared Selmer's and Stäckel's approximations g_{Se1} , g_{Se2} and g_{St} in the ranges $[6, 10^6]$ and $[5 \cdot 10^8, 2N]$. The corresponding values of $g_{Se1}(n)$ have been determined by numerical integration of

$$\int_2^{n-2} \frac{du}{\log u \log(n-u)}$$

instead of its asymptotical equivalent given above (see also [33], page 6).

The computation was done by using Mathematica's `NIntegrate`- routine, double checked by a second program. Tables 4 and 5 list the resulting values as given in Table 3, but restricted to the above ranges, this time including Selmer's and Stäckel's (using exact π - values) approximations. Except for the real errors, all values are in percentages.

In Table 5, Δh_X abbreviates the terms $h_X(2N) - h_X(5 \cdot 10^8 - 2)$.

4.2 Discussion

The relatively bad approximation of g by taking $1/\log^2 n$ as $P(n)$ as in Hardy/-Littlewood's formula seems to be due to that although asymptotically being equal, $\pi(n)$ is not very accurately described by $n/\log n$. But neither Stäckel's formula, using exact values of $\pi(n)$, nor the approximation $g_{R(n)}$ give as accurate values as Selmer's do. His second suggestion g_{Se2} yields very good results. It could be worth to also consider the next term of Riemann's π -formula in Selmer's integral, though this does again increase the already high computing costs of his estimation. Under the "easy computable" formulas, it was a bit surprising that Shah and Wilson's g_{SW1} gave values superior to those given by $g_{R(n)}$ in all ranges.

As for the method used, a decrease of the space needed by Algorithm 2 was essential in order to be able at all to compute $g(n)$ up to our limit of $5 \cdot 10^8$. Depending on the development of the memory/processor speed ratio in the future, this might well change. But it seems likely that using a greater number of small machines will still be cheaper for a longer time than incorporating supercomputers with very limited time access.

Remark

As kindly communicated to the author, Y. Saouter [32] has recently announced another way to compute values of the function g . Though the algorithm does extend the memory requirements of Algorithm 2, it promises a substantial decrease of computing time.

Acknowledgements

The author is indebted to the referees for their comments and for pointing out some corrections. Thanks also go to the staff of the Institut für Informatik, Universität Giessen for their support and the computing time necessary to carry out this work.

References

1. V. Aubry, *Vérification du théorème de Goldbach*. L'intermédiaire de math. **3** (1896) 75., **4** (1897), 60, **10** (1903), 61+62, (errata, 283).

2. C. Bays, R. Hudson, *The Segmented Sieve of Eratosthenes and Primes in Arithmetic Progressions*, BIT **17** (1977), 121–127.
3. J. Bohman, C. E. Fröberg, *Numerical Results on the Goldbach Conjecture*, BIT **15** (1975), 239–243.
4. N. V. Bougaief, C. R. Acad. Sci. Paris Sér. I Math. **100** (1885), 1124.
5. R. P. Brent, *The first occurrence of large gaps between successive primes*, Math. Comp., **27** (1973), 959–963.
6. V. Brun, *Über das Goldbachsche Gesetz und die Anzahl der Primzahlpaare*, Archiv for Math. og Naturvidenskab **34** Nr. 8 (1915), 8–19.
7. V. Brun, *Le crible d’Eratosthène et le théorème de Goldbach*, C. R. Acad. Sci. Paris Sér. I Math. **168** (1919), 544–546.
8. V. Brun, *Untersuchungen über das Siebverfahren des Eratosthenes*, Jahresber. Deutsch. Math.-Verein. **33** (1924), 81–96.
9. G. Cantor, *Vérification jusqu’à 1000 du théorème de Goldbach*, Association Française pour l’Avancement des Sciences, Congrès de Caen (1894), 117–134.
10. G. Cantor, L’intermédiaire des math. **2** (1895), 179.
11. G. Cantor, L’intermédiaire de math. **10** (1903) 168.
12. J.-M. Deshouilliers, A. Granville, W. Narkiewicz, C. Pomerance, *An upper bound in Goldbach’s problem*, Math. Comp. **61** (1993), 209–213.
13. L. E. Dickson, *Goldbach’s Theorem* in: History of the theory of numbers, Washington (1919), 421–424.
14. H. Halberstam, H.-E. Richert, *Sieve Methods*, Academic Press (1974).
15. G. H. Hardy, J. E. Littlewood, *Note on Messrs Shah and Wilson’s paper entitled: ‘On an emirical formula connected with Goldbach’s Theorem’*, Proc. Cambr. Phil. Soc. **19** (1919), 245–254.
16. G. H. Hardy, *Goldbach’s Theorem*, Mat. Tidsskrift B (1922), 1–16.
17. G. H. Hardy, J. E. Littlewood, *Some problems of ‘partitio numerorum’; III: On the expression of a number as a sum of primes*, Acta. Math. **44** (1922), 32–39.
18. R. Haussner, *Über das Goldbachsche Gesetz*, Jahresber. Deutsch. Math.-Verein. **5** (1896), 62–66.
19. R. Haussner, *Tafeln für das Goldbachsche Gesetz*, Nova Acta. Abh. der Kaiserl. Leop.-Carol. Deutschen Akademie der Naturforscher (1897), Band LXXII, Nr. 1, 1–214.
20. R. Haussner, (errata L. Ripert), L’intermédiaire de math. **10** (1903) 168.
21. E. Landau, *Über die zahlentheoretische Funktion $\varphi(n)$ und ihre Beziehung zum Goldbachschen Satz*, Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl. II, (1900) 177–186.
22. D. Lavenier, Y. Saouter *The Goldbach Conjecture: Checking it and counting the Partitions*, submitted for publication, (1999).
23. F. J. E. Lionnet, Nouv. Ann. Math. Sér. 2 (1879), 356–360.
24. N. Pipping, *Über Zwillingsprimzahlen und Goldbachsche Spaltungen*, Comm. Phys.-Math. III, **2** (1926), 1–14.
25. N. Pipping, *Neue Tafeln für das Goldbachsche Gesetz nebst Berichtigungen zu den Haussnerschen Tafeln*, Comm. Phys.-Math. IV, **4** (1927), 1–27.
26. N. Pipping, *Über Goldbachsche Spaltungen großer Zahlen*, Comm. Phys.-Math. IV, **10** (1927), 1–16.
27. N. Pipping, *Die Goldbachschen Zahlen $G(x)$ für $x = 120072 - 120170$* , Comm. Phys.-Math. IV, **25** (1929).
28. N. Pipping, *Die Goldbachsche Vermutung und der Goldbach-Vinogradovsche Satz*, Acta Acad. Aboensis, Math. Phys. **11** (1938), 4–25.

29. N. Pipping, *Spaltung der geraden X für X = 60000 bis 99998*, Acta Acad. Aboensis, Math. Phys. **12** (1940), 1–25.
30. J. Richstein, *Verifying the Goldbach Conjecture up to 4×10^{14}* , to appear in Math. Comp.
31. L. Ripert, *Nombre pair somme de deux nombres premiers*, L'intermédiaire de math. **10** (1903), 166+167.
32. Y. Saouter, *Computations of Goldbach partitions up to 128×10^6 with fft*, submitted for publication (1999).
33. E. S. Selmer, *Eine neue hypothetische Formel für die Anzahl der Goldbachschen Spaltungen einer geraden Zahl, und eine numerische Kontrolle*, Archiv for Math. og Naturvidenskab **46** Nr. 1 (1942), 1–18.
34. N. M. Shah., B. M. Wilson, *Numerical data connected with Goldbach's theorem*, Proc. London Math. Soc. **18** (1920), viii.
35. N. M. Shah., B. M. Wilson, *On an empirical formula connected with Goldbach's theorem*, Proc. London Math. Soc. **19** (1920), 238–245.
36. P. Stäckel, *Über Goldbach's empirisches Theorem: Jede grade Zahl kann als Summe von zwei Primzahlen dargestellt werden*, Nachr. Akad. Wiss. Göttingen Math.-Phys. Kl. II, (1896), 292–299.
37. P. Stäckel, *Die Darstellung der geraden Zahlen als Summen von zwei Primzahlen*, Sitzungsber. Heidelb. Akad. Wiss. (1916) **10** (1916), 1–47.
38. P. Stäckel, *Die Lückenzahlen r-ter Stufe und die Darstellung der geraden Zahlen als Summen und Differenzen ungerader Primzahlen*, Sitzungsber. Heidelb. Akad. Wiss. I. Teil (1917) **15**, 1–52, II. Teil (1918) **2**, 1–48 , III. Teil (1918) **14**, 1–67.
39. M. L. Stein, P. R. Stein, *Tables of the number of binary decompositions of all even numbers $0 < 2n < 200,000$ into prime numbers and lucky numbers*, Los Alamos Technical Report LA-3106-v.1 (1964), 1–442.
40. M. L. Stein, P. R. Stein *Experimental results on additive 2 bases*, BIT **38** (1965), 427–434.
41. M. L. Stein, P. R. Stein, *New experimental results on the Goldbach conjecture*. Math. Mag. **38**, 72–80, 1965.
42. F. J. Studnicka, *Bemerkung über gerade Zahlen*, Casopis **26** (1897), 207–208.
43. J. J. Sylvester, *On the partition on an even number into two primes* (Abstract of a talk given on November 9, 1871, London Math. Soc.). Proc. London Math. Soc. **4** (1871), 4–6. See also: Messenger of Mathematics **1** (1872), 127+128.
44. J. J. Sylvester, *On the Goldbach-Euler theorem regarding prime numbers*. Nature **55** (1896/97), 196–197, 269.
45. W. Weinreich, *Die Zwillingsdarstellungen der durch sechs teilbaren geraden Zahlen, berechnet für den Bereich von 6 bis 16800*, unpublished (available at the Mathe-matisches Institut der Universität Heidelberg) (1917), 214 pages.

Numerical Verification of the Brumer-Stark Conjecture

Xavier-François Roblot¹ and Brett A. Tangedal²

¹ CICMA, Concordia University, Montréal, Québec, Canada
`roblot@cs.concordia.ca`

² College of Charleston, Charleston SC, 29424, USA
`tangedal@math.cofc.edu`

1 Introduction

The construction of group ring elements that annihilate the ideal class groups of totally complex abelian extensions of \mathbb{Q} is classical and goes back to work of Kummer and Stickelberger. A generalization to totally complex abelian extensions of totally real number fields was formulated by Brumer. Brumer's formulation fits into a more general framework known as the Brumer-Stark conjecture. We will verify this conjecture for a large number of examples belonging to an extended class of situations where the general status of the conjecture is still unknown. We assume throughout that k is a totally real basefield and K is a totally complex extension field, abelian over k . Let w_K denote the number of roots of unity in K , $m = [k : \mathbb{Q}]$, and $G = \text{Gal}(K/k)$. We also let $S = S(K/k) = \{\mathfrak{p}_\infty^{(1)}, \dots, \mathfrak{p}_\infty^{(m)}, \mathfrak{p}_1, \dots, \mathfrak{p}_t\}$, where $\mathfrak{p}_\infty^{(i)}$ denotes the archimedean prime corresponding to the i th embedding of k into \mathbb{R} , and $\mathfrak{p}_1, \dots, \mathfrak{p}_t$ are precisely the prime ideals in k that ramify in K . For each $\sigma \in G$, we define a corresponding partial zeta-function

$$\zeta_S(s, \sigma) = \sum_{\sigma_a = \sigma} \frac{1}{N\mathfrak{a}^s} \quad (1)$$

where the sum is over all integral ideals \mathfrak{a} of k relatively prime to the ramified primes $\mathfrak{p}_1, \dots, \mathfrak{p}_t$ and having the same Artin symbol $(K/k, \mathfrak{a}) = \sigma_a = \sigma$. The infinite sum on the right side of (1) converges only for $\Re(s) > 1$, but $\zeta_S(s, \sigma)$ has a meromorphic continuation to all of \mathbb{C} with exactly one (simple) pole at $s = 1$. In particular, $\zeta_S(s, \sigma)$ is analytic at $s = 0$, and based upon work of Klingen [K] and Siegel [S], we know that $\zeta_S(0, \sigma) \in \mathbb{Q}$. A more refined result, due independently to Deligne and Ribet [DR], Barsky [B], and Cassou-Noguès [CN], states that $w_K \zeta_S(0, \sigma) \in \mathbb{Z}$ for every $\sigma \in G$. Based upon this, the group ring element

$$\gamma = \gamma_{K/k} = w_K \sum_{\sigma \in G} \zeta_S(0, \sigma) \sigma^{-1}$$

lies in $\mathbb{Z}[G]$. Following Hayes [H1], we refer to γ as the Brumer element of the extension K/k . The “anti-units” of K , denoted by K^\times , are the elements $\alpha \in K^\times$

having absolute value one at all archimedean primes of K . Let \mathfrak{B} be an arbitrary fractional ideal in K . We may now state the

BRUMER-STARK CONJECTURE: *There exists an anti-unit $\alpha \in K^\circ$ such that $(\alpha) = \mathfrak{B}^\gamma$ and $K(\alpha^{1/w_K})$ is abelian over k .*

Brumer originally conjectured that γ annihilates the ideal class group of K (i.e. that \mathfrak{B}^γ is always principal). The additional feature that an anti-unit generator of the principal ideal \mathfrak{B}^γ can be found whose w_K th root generates an abelian extension over k is due to Stark.

Before describing our computations, we first give a brief summary of the present state of the conjecture. The Brumer-Stark conjecture has already been proved in the following cases.

- (i) If $k = \mathbb{Q}$, using Stickelberger's Theorem (see [T2], p. 109).
- (ii) If $[K : k] = 2$ [T1].
- (iii) If $G \cong \mathbb{Z}_2 \times \mathbb{Z}_2$ in general, and when G is of exponent 2 and has order $2^l > 4$, assuming K/k is a tame extension [Sa1].
- (iv) If the class number of K is one, since the conjecture is always true for principal ideals [T1].
- (v) If $|G| = 4$ and K/k is a sub-extension of a non-abelian Galois extension K/k_0 of degree 8 [T1].
- (vi) If K/k is a sub-extension of an abelian Galois extension K/k_0 , and the Brumer-Stark conjecture is already known to be true for K/k_0 ([Sa2],[H2]).

Wiles made very important progress towards proving Brumer's part of the conjecture in [W]. For each prime p , he formulated a sub-conjecture for the p -part of the ideal class group of K , and showed that Brumer's conjecture follows if the sub-conjecture can be proven for every prime p . Following Wiles, Greither [G] has identified a large class of “nice” extensions and has proved that in these extensions the Brumer element annihilates the p -part of the ideal class group of K for all odd primes p . Working under these same restrictions, Popescu [P] has used Greither's results to deduce Stark's part of the conjecture as well. The prime 2 presents special difficulties because all of these results rely upon the Main Conjecture of Iwasawa Theory in a crucial way. Based upon this summary, we can describe the first general class of situations still unproven. The smallest basefield k would be real quadratic by (i). Since 2 always divides the relative degree $[K : k]$, the smallest unproven case would be where $G \cong \mathbb{Z}_4$ by (ii) and (iii). Therefore $[K : \mathbb{Q}] = 8$, and we restrict ourselves to those fields K whose class number exceeds one (by (iv)) and where K is non-Galois over \mathbb{Q} (by (v) and (i) and (vi) combined). The suggestion that this particular class of situations be studied numerically was already made by Tate in 1981 ([T1], p. 15), but a serious computational study has only become feasible in recent years with the availability of packages such as PARI/GP [BBBCO] and KANT [DFKPRSW].

We present our computations according to the following plan. In section 2, we describe a simple method that produces an abundant supply of totally complex \mathbb{Z}_4 extensions over any totally real basefield. Section 3 contains our algorithm for computing the Brumer element γ , which is uniformly applicable over any totally

real basefield. Section 4 gives a description of the computations required to verify the Brumer-Stark conjecture. Finally, a detailed example is presented in section 5, and section 6 contains tables and comments summarizing our computations.

2 Generating \mathbb{Z}_4 Extensions

The following theorem appears in a paper of Nagell [N]. Let k be an arbitrary basefield. Nagell proves (Thm. 3, p. 351) that any cyclic \mathbb{Z}_4 extension over k can be generated by a root β of the form $\sqrt{b(1 + c^2 + \sqrt{1 + c^2})}$ where $b, c \in k$. For our purposes, we assume k is a totally real number field and we can ensure that $K = k(\beta)$ is totally complex by choosing $b = -1$. The quartic polynomial that $\sqrt{-(1 + c^2 + \sqrt{1 + c^2})}$ satisfies is

$$f(x) = x^4 + 2(1 + c^2)x^2 + c^2(1 + c^2). \quad (2)$$

Let $c \in \mathcal{O}_k$, and assume that $1 + c^2 \notin k^2$. Then $f(x)$ will be irreducible over $k[x]$ by Theorem 2 of [KW] and any root of $f(x)$ will generate a totally complex \mathbb{Z}_4 extension K over k (see Thm. 3(ii) of [KW]).

3 Computation of the Brumer Element

With a relative extension K/k defined by an irreducible polynomial $f(x)$ as in section 2, one can use the tools of a computer package (we used PARI/GP) to compute the number of roots of unity w_K , and the relative discriminant $\mathfrak{d}(K/k)$. The primes $\mathfrak{p}_1, \dots, \mathfrak{p}_t$ dividing $\mathfrak{d}(K/k)$ are exactly the finite primes appearing in the set S . The only real task remaining in the computation of the Brumer element is the calculation of $\zeta_S(0, \sigma)$. Given $\sigma \in G$, we need a nice characterization of all integral ideals \mathfrak{a} relatively prime to $\mathfrak{d}(K/k)$ which have Artin symbol $\sigma_{\mathfrak{a}} = \sigma$. A beautiful characterization is provided by the Artin Reciprocity Law [Ha] which is most elegantly formulated in terms of the conductor $\mathfrak{f}(K/k)$ of the extension K/k . The conductor of a totally complex abelian extension of a totally real number field has the form

$$\mathfrak{f}(K/k) = \mathfrak{f} \mathfrak{p}_{\infty}^{(1)} \cdots \mathfrak{p}_{\infty}^{(m)},$$

where \mathfrak{f} is an integral ideal in k which has the exact same prime divisors as $\mathfrak{d}(K/k)$. With respect to the modulus $\mathfrak{f}(K/k)$, we obtain a partition of all fractional ideals in k relatively prime to \mathfrak{f} into a finite number of classes. These classes form an abelian group, called the ray class group mod $\mathfrak{f}(K/k)$, and denoted by $G(\mathfrak{f}(K/k))$. In general, several classes will correspond to a single automorphism $\sigma \in G$ via the Artin map, and $\zeta_S(s, \sigma)$ is formed by summing over exactly the integral ideals in these classes. Even with this problem solved, we still need to analytically continue $\zeta_S(s, \sigma)$ in order to compute it at $s = 0$. The best known method to date is to decompose $\zeta_S(s, \sigma)$ into a finite sum of “sector zeta-functions” and find an analytic continuation of these latter functions. Shintani

[Sh] accomplished this over any totally real basefield and found an explicit evaluation of the sector zeta-functions at $s = 0$ in terms of Bernoulli polynomials. The resulting formulas, as they stand, are impractical from an algorithmic point of view. On the other hand, one can use Shintani's evaluations in conjunction with a geometric method involving "convexity polygons" to obtain an efficient algorithm over a real quadratic basefield [H1]. This method can be generalized to any totally real basefield k of degree m over \mathbb{Q} by taking the convex closure of a set of lattice points in \mathbb{R}^m . Because of the resulting geometric complications, this method already has serious problems from an algorithmic standpoint when $m = 3$ (see [Kh], p. 276).

We use an alternate method which relies upon the decomposition of $\zeta_S(s, \sigma)$ into a sum of L -functions. The analytic continuation of the latter type of function is classical and dates back to Hecke. Recently, a very efficient method to compute L -function values has been used to test a related conjecture of Stark ([DST],[Ro]). We use this method here, which is based upon a formula due independently to Lavrik [L] and Friedman [F]. The relevant L -functions are defined from characters $\chi : G(\mathfrak{f}(K/k)) \rightarrow \mathbb{C}^\times$ on the ray class group mod $\mathfrak{f}(K/k)$. A given character will have a conductor of the form $\mathfrak{f}(\chi) = \mathfrak{f}_\chi \mathfrak{f}_{\chi, \infty}$, where \mathfrak{f}_χ is an integral ideal dividing \mathfrak{f} , and $\mathfrak{f}_{\chi, \infty}$ is a formal product of archimedean primes taken from the set $\{\mathfrak{p}_\infty^{(1)}, \dots, \mathfrak{p}_\infty^{(m)}\}$. We always work with the *primitive* version of χ (still denoted by χ), which is defined on the ray class group mod $\mathfrak{f}(\chi)$. The corresponding L -function is defined by

$$L(s, \chi) = \prod \left(1 - \frac{\chi(\mathfrak{p})}{N\mathfrak{p}^s} \right)^{-1} \quad \text{for } \Re(s) > 1,$$

with the product taken over all prime ideals in k relatively prime to \mathfrak{f}_χ . If we multiply $L(s, \chi)$ by the Euler factors corresponding to primes that divide \mathfrak{f} but not \mathfrak{f}_χ (there are potentially no such primes), we obtain a related function denoted by $L_S(s, \chi)$. Class field theory gives a correspondence (discussed in greater detail below) between a particular subset X_K of characters on $G(\mathfrak{f}(K/k))$ and the abelian extension K/k . In fact, the characters in X_K form a group that is isomorphic to $\text{Gal}(K/k)$. The decomposition of $\zeta_S(s, \sigma)$ into a sum of L -functions mentioned above is

$$\zeta_S(s, \sigma) = \frac{1}{[K : k]} \sum_{\chi \in X_K} \bar{\chi}(\mathfrak{a}) L_S(s, \chi), \tag{3}$$

where \mathfrak{a} is any ideal relatively prime to \mathfrak{f} such that $\sigma_{\mathfrak{a}} = \sigma$. All of the L -functions defined above have meromorphic continuations to the whole complex plane and all are analytic in particular at $s = 0$. Let $r(\chi)$ denote the order of the zero of the function $L(s, \chi)$ at $s = 0$.

Proposition 1. *Assume $[k : \mathbb{Q}] = m \geq 2$. For a given character χ defined on $G(\mathfrak{f}(K/k))$, we have $r(\chi) = 0$ if and only if $\mathfrak{f}_{\chi, \infty} = \mathfrak{p}_\infty^{(1)} \cdots \mathfrak{p}_\infty^{(m)}$.*

Proof. We refer to [DT] for the basic facts used here. For the trivial character, $\mathfrak{f}_{\chi_0, \infty} = 1$ and $r(\chi_0) = m - 1$ which is greater than 0 by assumption. If χ is

non-trivial, then $r(\chi) = m - q$, where q is the number of archimedean primes in the formal product $\mathfrak{f}_{\chi, \infty}$.

If $L(0, \chi) = 0$ for a given χ , then clearly $L_S(0, \chi) = 0$ as well. We will have $L(0, \chi) = 0$ for at least half of the characters in X_K and so the following restriction is important. Let $X_{K, \infty}$ denote the subset of characters $\chi \in X_K$ such that $\mathfrak{f}_{\chi, \infty} = \mathfrak{p}_\infty^{(1)} \cdots \mathfrak{p}_\infty^{(m)}$. Then equation (3) specializes to

$$\zeta_S(0, \sigma) = \frac{1}{[K : k]} \sum_{\chi \in X_{K, \infty}} \overline{\chi}(\mathfrak{a}) L_S(0, \chi). \quad (4)$$

In the following discussion, we focus on a fixed character $\chi \in X_{K, \infty}$. Before we can give the Lavrik-Friedman formula for the non-zero complex number $L(0, \chi)$, we need a few preliminary definitions. Let $A_\chi = \sqrt{d_k N \mathfrak{f}_\chi / \pi^m}$, where d_k is the discriminant of the field k and $N \mathfrak{f}_\chi$ is the norm of the integral ideal \mathfrak{f}_χ . Let $a_n(\chi)$ denote the finite sum $\sum \chi(\mathfrak{a})$ taken over all integral ideals \mathfrak{a} of norm n that are relatively prime to \mathfrak{f}_χ . Finally, let

$$f(x, s) = \frac{1}{2\pi i} \int_{\delta-i\infty}^{\delta+i\infty} x^z \left(\Gamma \left(\frac{z+1}{2} \right) \right)^m \frac{dz}{z-s}$$

for any $\delta > 1$. Then (see equations (4) and (5) of [DT])

$$L(0, \chi) = \frac{1}{\sqrt{\pi^m}} \sum_{n \geq 1} \left[a_n(\chi) f \left(\frac{A_\chi}{n}, 0 \right) + W(\chi) \overline{a_n(\chi)} f \left(\frac{A_\chi}{n}, 1 \right) \right], \quad (5)$$

where $W(\chi)$ is a complex number of absolute value one known as the “Artin root number” of χ . Let Y_χ denote the set of primes dividing \mathfrak{f} but not \mathfrak{f}_χ . If there exists a prime $\mathfrak{p} \in Y_\chi$ such that $\chi(\mathfrak{p}) = 1$, then $L_S(0, \chi) = 0$. Otherwise

$$L_S(0, \chi) = L(0, \chi) \prod_{\mathfrak{p} \in Y_\chi} (1 - \chi(\mathfrak{p})) \neq 0.$$

We would like to point out that Louboutin [Lo] has arrived at an equivalent form of equation (5) independently and has used it to compute relative class numbers of CM-fields. He actually gives a formula for $L(1, \chi)$, but the two values are related by $L(0, \chi) = \pi^{-m/2} W(\chi) A_\chi L(1, \bar{\chi})$ via the functional equation. Using equation (5), we can approximate $L(0, \chi)$ to any desired degree of accuracy by taking enough terms. We refer to [Lo] for detailed error bounds. The root number $W(\chi)$ is a finite sum with $\phi(\mathfrak{f}_\chi)$ terms. We refer to [DT] and [Lo] for its computation.

The method we have described thus far to compute $\zeta_S(0, \sigma)$ is applicable to any totally complex abelian extension K over a totally real field k . Throughout the remainder of this section we will assume that $G \cong \mathbb{Z}_4$. Since $G \cong \mathbb{Z}_4$, there is a ray class group character χ of order 4 corresponding to K/k , whose conductor $\mathfrak{f}(\chi)$ is equal to $\mathfrak{f}(K/k)$. The conductor of the trivial character χ_0 is

of course 1. The conductors of the two characters $\chi_1 = \chi$ and $\chi_3 = \chi^3$ are equal since they are conjugate to each other. The conductor of the character $\chi_2 = \chi^2$ contains no archimedean primes and is equal to the relative discriminant of the relative quadratic extension $k' = k(\sqrt{1+c^2})/k$. The conductor-discriminant formula [Ha] gives us the relation $\mathfrak{d}(K/k) = \mathfrak{d}(k'/k)\mathfrak{f}^2$ and thus an immediate determination of \mathfrak{f} . We can now compute $G(\mathfrak{f}(K/k))$, but we still have to identify the exact set of characters $X_K = \{\chi_0, \chi_1, \chi_2, \chi_3\}$ corresponding to the extension K/k . To do this, we need to generate a subgroup of index 4 using relative norms. Based upon the following result of Bach and Sorenson [BS] (which assumes the ERH), we don't have to work too hard. Let

$$C = (4 \log |d_K| + 2.5[K : \mathbb{Q}] + 5)^2$$

and let T denote the set of prime ideals in k of degree 1 over \mathbb{Q} not dividing \mathfrak{f} and having norm $\leq C$. Let H be the subgroup in $G(\mathfrak{f}(K/k))$ generated by the ideals $\mathfrak{p}^{f_{\mathfrak{p}}}$, where \mathfrak{p} runs through T and $f_{\mathfrak{p}}$ denotes the residue degree of \mathfrak{p} in K/k . Then $[G(\mathfrak{f}(K/k)) : H] = 4$ and the four characters on $G(\mathfrak{f}(K/k))$ which are trivial on H make up the set X_K . We have $L(s, \chi_i) = L_S(s, \chi_i)$ for $i = 1, 3$ and $L_S(0, \chi_i) = 0$ for $i = 0, 2$. Since $L(0, \chi_1) = \overline{L(0, \chi_3)}$, we finally obtain

$$w_K \zeta_S(0, \sigma) = \frac{w_K}{2} (\Re(\overline{\chi_1}(\mathfrak{a}) L(0, \chi_1))) \quad (6)$$

from equation (4). Recalling that $w_K \zeta_S(0, \sigma) \in \mathbb{Z}$, we just need to compute $L(0, \chi_1)$ to high enough accuracy to determine the *integer* on the right side of (6).

Even though we haven't made a detailed comparison between our method and the method in [H1], we believe that our method performs equally well when $m = 2$ and certainly much better when $m > 2$.

4 The Numerical Verification of the Conjecture

We begin with

Proposition 2. *Let $\mathfrak{B}_1, \dots, \mathfrak{B}_s$ be a system of $\mathbb{Z}[G]$ -generators of the ideal class group of K . Then the Brumer-Stark conjecture is true for every fractional ideal \mathfrak{B} in K if and only if it is true for each ideal \mathfrak{B}_i with $1 \leq i \leq s$.*

Proof. This is a direct consequence of the properties of the subgroup of fractional ideals verifying the Brumer-Stark conjecture (see [T1], p. 7).

Thus, it is enough to verify the conjecture for a finite set of ideals \mathfrak{B}_i , $1 \leq i \leq s$. Furthermore, a system of $\mathbb{Z}[G]$ -generators can easily be extracted from a system of ideals that generate the ideal class group over \mathbb{Z} .

To prove that a given fractional ideal \mathfrak{B} in K verifies the Brumer-Stark conjecture we proceed as follows. We first compute the ideal \mathfrak{B}^γ and its class in the class group. If it is not principal, then the conjecture is false. Otherwise, we

compute a generator β of \mathfrak{B}^γ . In general, this generator is not an anti-unit and requires modification. If an anti-unit generator α exists, then we have $\alpha = \varepsilon\beta$ for some unit ε of K . The unit ε is determined using the process described below. We assume that $G \cong \mathbb{Z}_4 = \langle \sigma \rangle$ and $[k : \mathbb{Q}] = 2$ throughout the remainder of this section.

Let $K \hookrightarrow K^{(i)} \subset \mathbb{C}$, $i = 1, 2, 3, 4$ be four non complex conjugate embeddings. For $\alpha \in K$, let $|\alpha|_i = |\alpha^{(i)}|^2$ be the normalized absolute value. Consider the classical logarithmic embedding

$$\begin{aligned}\lambda : K^\times &\rightarrow \mathbb{R}^3 \\ \alpha &\mapsto (\log |\alpha|_1, \log |\alpha|_2, \log |\alpha|_3).\end{aligned}$$

The anti-units are contained in the kernel of λ , and if α exists then

$$\lambda(\varepsilon) + \lambda(\beta) = 0.$$

Let $\|\cdot\|$ be the Euclidean norm on \mathbb{R}^3 and let b be the minimal non-zero norm $\|\lambda(u)\|$ where u ranges through the units of K . Then the unit ε , if it exists, is the unique unit (up to some root of unity in K) satisfying

$$\|\lambda(\varepsilon) + \lambda(\beta)\| < b/2.$$

This unit can be found using computation with real numbers, however once it is found, we still need to check that α possesses the required properties. The following proposition allows us to verify that α is an anti-unit.

Proposition 3. $\alpha \in K^\circ$ if and only if $\alpha^{1+\sigma^2} = 1$.

Proof. The automorphism σ^2 is the unique complex conjugation of the extension K/k , thus $|\alpha|_i = |\alpha^{\sigma^2}|_i$ for all i 's, so $|\alpha^{1+\sigma^2}|_i = |\alpha|_i^2$ and also $\alpha^{1+\sigma^2}$ is a positive real number. Now assume that α is an anti-unit. Then $|\alpha|_i = 1$ for all i 's, and thus $\alpha^{1+\sigma^2} = 1$. On the other hand, if $\alpha^{1+\sigma^2} = 1$, then $|\alpha|_i^2 = 1$ for all i 's and α is an anti-unit.

Since ε is unique up to a root of unity in K , so is α . Therefore, the condition that $K(\alpha^{1/w_K})$ generates an abelian extension over k does not depend upon the choice of α . The next proposition allows us to verify this condition. We first note that $w_K = 2$ for all of the fields suggested by Tate for study (see Introduction). To see this, let L be the field generated over \mathbb{Q} by the roots of unity contained in K . If $w_K > 2$, then Lk is a totally complex sub-extension of K/k . Therefore $Lk = K$ and K/\mathbb{Q} is abelian, which gives a contradiction.

Proposition 4. $K(\sqrt{\alpha})$ is abelian over k if and only if $\alpha^{\sigma-1} \in K^2$.

Proof. This follows directly from Prop. 1.2, p. 83 of [T2].

Note that all of the required computations are done with exact objects and therefore give a complete verification of the Brumer-Stark conjecture for all

of the examples tested and not just a verification up to the precision of the computation!

Since the prime 2 seems to play a special role in the conjecture, we make the following definition. We call the maximum power of 2 that can be factored out of the Brumer element γ the “2-part of γ ”. We have actually tested the conjecture in such a way as to see how much of the 2-part of γ is really needed. More precisely, let 2^e be the 2-part of γ . We have searched for the smallest non-negative integer i such that the conjecture is true with γ replaced by $2^{i-e}\gamma$. These results are described in the last section.

5 An Example

Let $k = \mathbb{Q}(\sqrt{2})$ and let K be the field generated by the polynomial (2) with $c = 3 + 3\sqrt{2}$. The discriminant d_K is $2^{31} \cdot 17^3$ and the conductor $\mathfrak{f}(K/k)$ is $\mathfrak{p}_2^7 \mathfrak{p}_{17} \mathfrak{p}_\infty^{(1)} \mathfrak{p}_\infty^{(2)}$, where $\mathfrak{p}_2 = (\sqrt{2})$ is the unique prime ideal above 2 and $\mathfrak{p}_{17} = (1 + 3\sqrt{2})$ is one of the two prime ideals above 17. The field K is generated over \mathbb{Q} by an algebraic integer θ satisfying

$$\theta^8 + 40\theta^6 + 380\theta^4 + 1360\theta^2 + 1666 = 0.$$

The field K is not Galois over \mathbb{Q} and its class group is isomorphic to $\mathbb{Z}_{20} \times \mathbb{Z}_2$. Moreover, its class group is generated over $\mathbb{Z}[G]$ by the class of the ideal

$$\mathfrak{B}_1 = 3\mathcal{O}_K + (\theta + 1)\mathcal{O}_K.$$

The Galois group G of the extension K/k is generated by the automorphism

$$\sigma : \theta \mapsto \frac{1}{567} (5\theta^7 + 235\theta^5 + 2978\theta^3 + 8935\theta).$$

We compute the Brumer element and find that

$$\gamma = 8 - 16\sigma - 8\sigma^2 + 16\sigma^3 = 2^3 (1 - 2\sigma - \sigma^2 + 2\sigma^3).$$

We start by testing $\gamma/8$, but the ideal $\mathfrak{B}_1^{\gamma/8}$ is not principal. Next, we look at the ideal $\mathfrak{B}_1^{\gamma/4}$ which is principal, and using the method described in the previous section we find that it is generated by the anti-unit

$$\alpha = \frac{1}{5103} (110\theta^7 + 98\theta^6 + 4036\theta^5 + 3724\theta^4 + 28346\theta^3 + 29288\theta^2 + 53056\theta + 63679).$$

However, the algebraic number $\alpha^{\sigma-1}$ is not a square in K , so the condition in Proposition 4 is not satisfied. Finally, it is clear that all the conditions are satisfied for $\gamma/2$.

Theorem 1. *The Brumer-Stark conjecture is true for this extension, it is even true if one replaces the Brumer element γ by $\gamma/2$.*

6 Tables and Summary

We used the quartic polynomial (2) with k ranging through the real quadratic fields of discriminant ≤ 500 and c ranging through the algebraic integers in k of T_2 -norm ≤ 200 with $1 + c^2 \notin k^2$. Discarding the fields K obtained in this way that have class number one, are Galois over \mathbb{Q} , or have discriminant $\geq 10^{18}$, and keeping only non-isomorphic fields, we end up with a list of 379 fields. The Brumer-Stark conjecture has been tested for each of these field extensions using the package PARI/GP [BBBCO].

Theorem 2. *The Brumer-Stark conjecture is true for all 379 field extensions listed in the tables below.*

In the following tables, we list the discriminants d of the real quadratic fields considered and the corresponding elements c . We set $\omega = (1 + \sqrt{d})/2$ if $d \equiv 1 \pmod{4}$ and $\omega = \sqrt{d}/2$ if $d \equiv 0 \pmod{4}$. The ring of integers of the real quadratic field k of discriminant d is $\mathcal{O}_k = \mathbb{Z} + \mathbb{Z}\omega$.

d	Values of c
5	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 2\omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 8 + 2\omega, -1 + 3\omega, 3\omega, 1 + 3\omega, 2 + 3\omega, 3 + 3\omega, 4 + 3\omega, 5 + 3\omega, 6 + 3\omega, -1 + 4\omega, 4\omega, 1 + 4\omega, 2 + 4\omega, 3 + 4\omega, 4 + 4\omega, 5 + 4\omega, 6 + 4\omega, -2 + 5\omega, -1 + 5\omega, 5\omega, 1 + 5\omega, 2 + 5\omega, 3 + 5\omega, 4 + 5\omega, 5 + 5\omega, -2 + 6\omega, -1 + 6\omega, 6\omega, 1 + 6\omega, 2 + 6\omega, 3 + 6\omega, 4 + 6\omega, 7\omega, 1 + 7\omega, 2 + 7\omega, -2 + 8\omega, 8\omega$
8	$2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 7 + \omega, 8 + \omega, 1 + 2\omega, 2 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega, 8 + 2\omega, 9 + 2\omega, 1 + 3\omega, 2 + 3\omega, 3 + 3\omega, 4 + 3\omega, 5 + 3\omega, 6 + 3\omega, 7 + 3\omega, 8 + 3\omega, 9 + 3\omega, 1 + 4\omega, 2 + 4\omega, 3 + 4\omega, 4 + 4\omega, 5 + 4\omega, 6 + 4\omega, 7 + 4\omega, 8 + 4\omega, 1 + 5\omega, 2 + 5\omega, 3 + 5\omega, 4 + 5\omega, 5 + 5\omega, 6 + 5\omega, 1 + 6\omega, 2 + 6\omega, 3 + 6\omega, 4 + 6\omega, 5 + 6\omega$
12	$3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 7 + \omega, 8 + \omega, 9 + \omega, 1 + 2\omega, 2 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega, 8 + 2\omega, 1 + 3\omega, 2 + 3\omega, 3 + 3\omega, 4 + 3\omega, 5 + 3\omega, 6 + 3\omega, 7 + 3\omega, 8 + 3\omega, 1 + 4\omega, 2 + 4\omega, 3 + 4\omega, 4 + 4\omega, 5 + 4\omega, 6 + 4\omega, 7 + 4\omega, 1 + 5\omega, 2 + 5\omega, 3 + 5\omega, 4 + 5\omega, 5 + 5\omega$
13	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 8 + \omega, 2\omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 3\omega, 1 + 3\omega, 2 + 3\omega, 3 + 3\omega, 4 + 3\omega, 5 + 3\omega, 2 + 4\omega, 4 + 4\omega$
17	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 2\omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 8 + 2\omega, 2 + 3\omega, 3 + 3\omega, 4 + 3\omega, 5 + 3\omega, 6 + 3\omega, 2 + 4\omega$
21	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 2\omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 3\omega, 5 + 3\omega, 2 + 4\omega$
24	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 9 + \omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega, 8 + 2\omega, 5 + 3\omega, 6 + 3\omega$
28	$1 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 8 + \omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega, 8 + 2\omega, 2 + 3\omega, 4 + 3\omega, 5 + 3\omega, 6 + 3\omega$
29	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 6 + \omega, 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 3 + 3\omega$
33	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega$
37	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega$
40	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 1 + 2\omega, 2 + 2\omega, 3 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega, 3 + 3\omega$
41	$\omega, 1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 7 + \omega, 2 + 2\omega, 4 + 2\omega, 6 + 2\omega$

d	Values of c	d	Values of c
44	$2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega,$ $8 + \omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega, 7 + 2\omega$	53	$2 + \omega, 3 + \omega, 4 + \omega$
56	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega,$ $7 + \omega, 2 + 2\omega, 4 + 2\omega, 5 + 2\omega, 6 + 2\omega$	57	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega$
60	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega,$ $6 + \omega, 8 + \omega$	61	$2 + \omega, 3 + \omega, 4 + \omega$
65	$2 + \omega, 3 + \omega, 4 + \omega$	69	$\omega, 2 + \omega, 3 + \omega, 4 + \omega, 2 + 2\omega$
73	$3 + \omega, 4 + \omega$	76	$1 + \omega, 2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega,$ $6 + \omega, 8 + \omega, 1 + 2\omega$
77	$4 + \omega, 7 + \omega$	77	$4 + \omega, 74 + \omega$
88	$3 + \omega, 4 + \omega, 5 + \omega$	89	$1 + \omega, 4 + \omega$
92	$2 + \omega, 3 + \omega, 4 + \omega, 5 + \omega, 6 + \omega,$ $8 + \omega$	101	$6 + \omega$
104	$4 + \omega, 5 + \omega, 6 + \omega, 8 + \omega$	109	$5 + \omega$
113	$3 + \omega, 4 + \omega$	120	$4 + \omega, 5 + \omega$
124	$2 + \omega, 4 + \omega, 5 + \omega, 6 + \omega, 8 + \omega$	129	$4 + \omega, 6 + \omega$
136	$4 + \omega$	137	$3 + \omega$
140	$4 + \omega, 6 + \omega, 8 + \omega$	141	$3 + \omega, 7 + \omega$
149	$4 + \omega$	156	$1 + \omega, 4 + \omega, 6 + \omega$
161	ω	172	$4 + \omega, 6 + \omega$
184	$7 + \omega$	188	$4 + \omega, 6 + \omega$
201	$6 + \omega$	204	$6 + \omega$
236	$4 + \omega$	237	$4 + \omega$
284	$4 + \omega$	321	$1 + \omega$

We now give some insight into how much of the 2-part of the Brumer element is needed for the conjecture to be true (see the comment at the end of section 4). First note that in all of our examples the Brumer element had a non-trivial 2-part. This is not generally true (see example 1, p. 172 of [H1]), but it might be true for certain classes of situations. More precisely, we have 3 examples (0.8%) for which the 2-part is 2, 207 examples (54.6%) for which it is 2^2 , 123 examples (32.4%) for which it is 2^3 , 40 examples (10.6%) for which it is 2^4 and 6 examples (1.6%) for which it is 2^5 . In all examples, the full 2-part is not needed for the conjecture to be true. Even more striking is that in 324 examples (85.5%) only half or less than half of the 2-part is necessary and in 96 examples (25.3%) the full 2-part can be removed. The value of 2^i (i.e. the part of the 2-part needed for the conjecture to be valid) was 1 for 96 examples (25.3%), 2 for 204 examples (53.8%), 2^2 for 67 examples (17.7%), 2^3 for 11 examples (2.9%) and 2^4 for 1 example (0.3%). The values of 2^{e-i} (i.e. the maximal part of the 2-part that can be removed) was 2 for 173 examples (45.7%), 2^2 for 190 examples (50.1%) and 2^3 for 16 examples (4.2%).

The following tables list the ideal class groups of all fields K considered. Each entry consists of two parts. The first part gives the invariant factor decomposition of an abelian group A in the form (n_1, \dots, n_r) where $n_j \geq 2$ for all j and $n_{i+1} | n_i$ for $1 \leq i < r$. The group A has structure $\mathbb{Z}_{n_1} \times \dots \times \mathbb{Z}_{n_r}$. The second part gives the

number of class groups isomorphic to A . Note that the smallest class number was 2 and the largest was 10064.

Occurrence of class groups with 1 invariant factor

(2)	9	(5)	2	(10)	17	(26)	4	(34)	4	(50)	3
(52)	2	(58)	3	(74)	2	(82)	2	(106)	2	(113)	1
(122)	1	(130)	5	(136)	1	(146)	1	(148)	1	(170)	1
(178)	1	(194)	1	(202)	1	(212)	1	(226)	1	(250)	1
(274)	1	(338)	1	(340)	1	(346)	1	(388)	1	(410)	1
(466)	1	(530)	1	(562)	1	(650)	1	(692)	1	(794)	1
(1130)	1	(1604)	1	(1810)	1	(1930)	1	(2026)	1	(2722)	1
(5910)	1										

Occurrence of class groups with 2 invariant factors

(2, 2)	4	(4, 2)	3	(4, 4)	3	(6, 6)	4	(8, 4)	1	(10, 2)	13
(10, 5)	1	(10, 10)	1	(12, 6)	1	(12, 12)	1	(16, 8)	1	(20, 2)	5
(20, 4)	3	(20, 10)	3	(22, 22)	1	(24, 12)	1	(26, 2)	3	(26, 13)	1
(28, 14)	1	(30, 3)	1	(30, 30)	1	(34, 2)	4	(36, 18)	1	(40, 4)	1
(50, 2)	3	(50, 10)	1	(52, 2)	2	(52, 4)	2	(52, 13)	1	(58, 2)	1
(60, 6)	1	(68, 2)	1	(70, 7)	1	(70, 14)	1	(72, 9)	1	(74, 2)	2
(78, 3)	2	(82, 2)	5	(100, 2)	4	(100, 4)	1	(100, 10)	1	(102, 3)	1
(106, 2)	1	(116, 2)	4	(130, 2)	4	(130, 10)	2	(146, 2)	1	(148, 2)	2
(150, 3)	1	(156, 6)	1	(164, 2)	3	(170, 2)	1	(178, 2)	4	(200, 8)	1
(204, 2)	1	(212, 2)	1	(218, 2)	3	(226, 2)	1	(232, 2)	1	(244, 2)	2
(260, 2)	2	(296, 2)	1	(300, 6)	1	(338, 2)	2	(340, 2)	5	(346, 2)	1
(356, 2)	1	(370, 2)	1	(390, 2)	1	(390, 3)	1	(404, 2)	2	(410, 2)	2
(424, 2)	1	(452, 2)	1	(482, 2)	1	(488, 2)	1	(500, 2)	1	(530, 2)	1
(580, 2)	1	(580, 4)	1	(596, 2)	1	(628, 2)	1	(772, 2)	1	(820, 2)	1
(822, 2)	1	(984, 4)	1	(1096, 2)	1	(1172, 2)	2	(1220, 2)	1	(2180, 2)	1

Occurrence of class groups with 3 invariant factors

(4, 2, 2)	2	(4, 4, 2)	3	(4, 4, 4)	2	(8, 4, 2)	3	(10, 2, 2)	3
(10, 10, 2)	1	(12, 6, 2)	2	(16, 8, 2)	1	(16, 8, 4)	1	(16, 8, 8)	1
(18, 18, 2)	1	(20, 2, 2)	10	(20, 4, 2)	5	(20, 4, 4)	1	(20, 10, 2)	2
(20, 20, 2)	3	(20, 20, 4)	1	(24, 12, 2)	1	(26, 2, 2)	3	(28, 14, 2)	3
(34, 2, 2)	3	(40, 2, 2)	1	(40, 4, 2)	1	(40, 4, 4)	1	(40, 8, 2)	1
(40, 20, 2)	1	(44, 44, 2)	1	(50, 2, 2)	2	(52, 2, 2)	3	(52, 4, 2)	2
(58, 2, 2)	1	(60, 12, 2)	1	(60, 12, 6)	1	(68, 2, 2)	4	(68, 4, 2)	1
(100, 2, 2)	1	(100, 4, 2)	2	(100, 10, 2)	1	(104, 2, 2)	2	(104, 4, 4)	1
(104, 8, 4)	1	(106, 2, 2)	2	(116, 2, 2)	3	(120, 12, 2)	1	(130, 2, 2)	1
(148, 2, 2)	2	(178, 2, 2)	1	(194, 2, 2)	1	(202, 2, 2)	1	(244, 2, 2)	1
(250, 2, 2)	1	(260, 2, 2)	1	(274, 2, 2)	1	(290, 2, 2)	1	(292, 4, 2)	1
(340, 2, 2)	2	(340, 4, 2)	1	(404, 2, 2)	1	(520, 2, 2)	1	(596, 2, 2)	1
(740, 2, 2)	1	(1830, 2, 2)	1	(2516, 2, 2)	1				

Occurrence of class groups with 4 invariant factors

(6, 6, 2, 2)	1	(8, 8, 2, 2)	2	(20, 2, 2, 2)	1	(20, 4, 2, 2)	1
(20, 4, 4, 2)	1	(20, 10, 2, 2)	1	(36, 36, 2, 2)	1	(52, 4, 2, 2)	1
(58, 2, 2, 2)	1	(68, 2, 2, 2)	1	(68, 4, 2, 2)	1	(82, 2, 2, 2)	1
(100, 4, 2, 2)	1	(104, 2, 2, 2)	1	(116, 4, 2, 2)	1	(122, 2, 2, 2)	1
(148, 2, 2, 2)	1	(200, 4, 2, 2)	1				

Occurrence of class groups with 5 invariant factors

$$(10, 2, 2, 2, 2) \parallel 1 \parallel (20, 2, 2, 2, 2) \parallel 1 \parallel (68, 4, 2, 2, 2) \parallel 1$$

Final note and acknowledgements. After having completed the full verification of the Brumer-Stark conjecture for all 379 examples listed here, Greither verified for us that all of our extensions are “nice” in the technical sense defined in his paper [G]. This makes our study of the 2-part of the Brumer element especially interesting (see comments at the end of section 1). We would like to thank Cornelius Greither for his help and we would also like to thank Igor Schein for helping us verify some of the most difficult examples.

References

- [BS] E. Bach and J. Sorenson: Explicit bounds for primes in residue classes. *Math. Comp.* **65** (1996) 1717–1735.
- [B] D. Barsky: Fonctions zêta p -adiques d’une classe de rayon des corps de nombres totalement réels. *Groupe d’étude d’analyse ultramétrique* 1977–78. Errata, idem 1978–79.
- [BBBCO] C. Batut, K. Belabas, D. Bernardi, H. Cohen, M. Olivier: User’s Guide to PARI/GP version 2.0.17, 1999.
- [CN] Pierrette Cassou-Noguès: Valeurs aux entiers négatifs des fonctions zêta et fonctions zêta p -adiques. *Invent. Math.* **51** (1979) 29–59.
- [DFKPRSW] M. Daberkow, C. Fieker, J. Klüners, M. Pohst, K. Roegner, M. Schörnig, K. Wildanger: KANT v.4. *J. Symb. Comput.* **24** (1997) 267–283.
- [DR] P. Deligne and K. Ribet: Values of abelian L -functions at negative integers over totally real fields. *Invent. Math.* **59** (1980) 227–286.
- [DST] David S. Dummit, Jonathan W. Sands, and Brett A. Tangedal: Computing Stark units for totally real cubic fields. *Math. Comp.* **66** (1997) 1239–1267.
- [DT] David S. Dummit and Brett A. Tangedal: Computing the lead term of an abelian L -function. ANTS III (Buhler, Ed.) LNCS Springer-Verlag, Berlin-Heidelberg-New York, **1423** (1998) 400–411.
- [F] Eduardo Friedman: Hecke’s integral formula. Séminaire de Théorie des Nombres Univ. Bordeaux I, Talence (1987–88) Exposé n° 5.
- [G] Cornelius Greither: Some cases of Brumer’s conjecture for abelian CM extensions of totally real fields. To appear in *Math. Zeit.*
- [Ha] Helmut Hasse: Vorlesungen über Klassenkörpertheorie. Physica-Verlag, Würzburg, 1967.
- [H1] David R. Hayes: Brumer elements over a real quadratic base field. *Expo. Math.* **8** (1990) 137–184.

- [H2] David R. Hayes: Base change for the conjecture of Brumer-Stark. *J. Reine Angew. Math.* **497** (1998) 83–89.
- [Kh] M. Khan: Computation of partial zeta values at $s = 0$ over a totally real cubic field. *J. Number Theory* **57** (1996) 242–277.
- [KW] L.-C. Kappe and B. Warren: An elementary test for the Galois group of a quartic polynomial. *Am. Math. Monthly* **96** (1989) 133–137.
- [K] H. Klingen: Über die Werte der Dedekindsche Zetafunktion. *Math. Ann.* **145** (1962) 265–272.
- [L] A. F. Lavrik: On functional equations of Dirichlet functions. English translation in *Math. USSR-Izvestija* **1** (1967) 421–432.
- [Lo] Stéphane Louboutin: Computation of relative class numbers of CM-fields by using Hecke L -functions. *Math. Comp.* **69** (2000) 371–393.
- [N] T. Nagell: Sur quelques questions dans la théorie des corps biquadratiques. *Arkiv för Matematik* **4** (1962) 347–376.
- [P] Cristian Popescu: E-mail communication received on August 12th, 1999. Paper(s) forthcoming.
- [Ro] Xavier-François Roblot: Algorithmes de factorisation dans les extensions relatives et applications de la conjecture de Stark à la construction des corps de classes de rayon. Thèse, Université Bordeaux I, 1997.
- [Sa1] Jonathan W. Sands: Galois groups of exponent two and the Brumer-Stark conjecture. *J. Reine Angew. Math.* **349** (1984) 129–135.
- [Sa2] Jonathan W. Sands: Abelian fields and the Brumer-Stark conjecture. *Comp. Math.* **53** (1984) 337–346.
- [Sh] T. Shintani: On evaluation of zeta functions of totally real algebraic number fields at non-positive integers. *J. Fac. Sci. Tokyo 1A* **23** (1976) 393–417.
- [S] C. L. Siegel: Über die Fourierschen Koeffizienten von Modulformen. *Nachr. Akad. Wiss. Göttingen* **3** (1970) 15–56.
- [T1] John Tate: Brumer-Stark-Stickelberger. Séminaire de Théorie des Nombres Univ. Bordeaux I, Talence (1980–81) Exposé n° 24.
- [T2] John Tate: Les Conjectures de Stark sur les Fonctions L d'Artin en $s = 0$. Progress in Math. Vol. 47, Birkhäuser, Boston, 1984.
- [W] Andrew Wiles: On a conjecture of Brumer. *Ann. Math.* **131** (1990) 555–565.

Explicit Models of Genus 2 Curves with Split CM

Fernando Rodriguez-Villegas*

University of Texas at Austin, TX 78712, USA
villegas@math.utexas.edu
<http://www.ma.utexas.edu/users/villegas>

Abstract. We outline a general algorithm for computing an explicit model over a number field of any curve of genus 2 whose (unpolarized) Jacobian is isomorphic to the product of two elliptic curves with CM by the same order in an imaginary quadratic field. We give the details and some examples for the case where the order has prime discriminant and class number one.

1 Motivation

Let E_1, E_2 be two elliptic curves defined over $\overline{\mathbb{Q}}$ with complex multiplication by an order \mathcal{O} of an imaginary quadratic field K . We are interested in finding explicit models for curves C defined over $\overline{\mathbb{Q}}$ whose (unpolarized) Jacobian is isomorphic to $E_1 \times E_2$. In this paper we propose a general algorithm for this purpose and give details only for the following special case where we have carried them out: $E_1 = E_2$, \mathcal{O} is the ring of integers of $K = \mathbb{Q}(\sqrt{-N})$, $N \equiv 3 \pmod{4}$ prime and \mathcal{O} has class number one. Our special case consists of finitely many curves, up to isomorphism; the algorithm produces models over K for them.

It is a general fact due to Narasimhan and Nori [NN] that there are only finitely many principal polarizations on a given abelian variety up to isomorphism. Hence, for a fixed \mathcal{O} there are only finitely many isomorphism classes of the curves we want; their number was calculated by Hayashida and Nishi [HN].

For a similar question in the case of abelian surfaces with complex multiplication by a quartic field see [vW].

Our interest in this problem arose in connection with a generalization to genus 2 of the *singular moduli* formulae of Gross and Zagier [GZ] for the norm of the difference of j -values of CM elliptic curves. (This generalization will be the subject of a separate publication.) As an illustration, consider the genus 2 curve C determined by

$$y^2 = f(x) = 6^{-3} h(x) h^\iota(x),$$

* Much of this work was done while I was a guest at the Max Planck Institut für Mathematik in Bonn in 1995. I take the opportunity to thank everybody at the Institut for their hospitality. I would also like to thank the NSF, TARP, and the Alfred P. Sloan Foundation for their generous support.

where

$$h(x) = (7144\sqrt{-163} - 151790)x^3 + (129789\sqrt{-163} + 1752597)x^2 + (-47481\sqrt{-163} + 510153)x + (-1596\sqrt{-163} - 37250),$$

and

$$h^\ell(x) = \overline{x^3 h(-1/x)}$$

(bar denoting complex conjugation of the coefficients). The unpolarized Jacobian of C is isomorphic over $\overline{\mathbb{Q}}$ to the product of two elliptic curves with CM by the ring of integers of $K = \mathbb{Q}(\sqrt{-163})$. Let

$$D = 2^{-12} \operatorname{disc}(f) = (2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23)^{12}.$$

Then we have

$$\log D = -6 \sum_{m \in \mathbb{Z}^3} \sum_{d|(163 - Q(m))/4} \left(\frac{-163}{d} \right) \log d,$$

where

$$Q(m) = m^t \begin{pmatrix} 24 & 4 & 6 \\ 4 & 55 & 1 \\ 6 & 1 & 83 \end{pmatrix} m$$

is a certain positive definite ternary quadratic form of level 163 associated to C and the (finite) sum is over $m \in \mathbb{Z}^3$ such that $(163 - Q(m))/4$ is a positive integer. In particular every rational prime l dividing D is smaller than $163/4$ and inert in K .

The significance of the number D is that C has bad reduction only at primes dividing D . Note that over $\overline{\mathbb{Q}}$ the Jacobian of C has good reduction everywhere but C does not; at primes dividing D , C reduces to two elliptic curves crossing at a point.

Another source of interest in the problem is the fact, which I learned from K. Lauter, that the reduction of the curves C provides genus 2 curves over certain finite fields with maximal number of rational points (see §5 for an example). In this regard, the more interesting problem is the analogous one for curves of genus 3 for which we hope to exhibit in the near future an algorithm similar to the one sketched here.

2 Outline of the Algorithm

We start by giving an outline of the main steps of the general algorithm and then give details for our special case in the next sections.

Step 1. Find period matrices for the polarized Jacobians.

Step 2. Given a non-split period matrix obtained in step 1 compute a model for the corresponding curve.

The first step is purely algebraic and only requires computations with rational numbers; it involves the calculation of representatives for ideal classes of certain orders in a quaternion algebra (see [HN] for more details). What we need to do is describe explicitly the finitely many principal polarizations on $E_1 \times E_2$ up to equivalence.

The second step relies on the following explicit version of Torelli's theorem for curves of genus 2 due to Bolza and Klein. Let

$$\mathcal{H}_2 = \{Z \in \mathbb{C}^{2 \times 2} \mid Z^t = Z, \quad \text{Im}(Z) \text{ positive definite}\}$$

be the Siegel upper-half space of rank 2. Let $Z \in \mathcal{H}_2$ be a period matrix of a principally polarized abelian surface which is not the product of two elliptic curves with the product polarization. A theorem of Torelli guarantees that Z arises from a curve of genus 2, unique up to isomorphism. Here is a way of recovering the curve from the period matrix Z .

Let $f_Z(u_1, u_2) \in \mathbb{C}[u_1, u_2]$ be the leading term in the Taylor expansion of

$$\prod_{(\mu, \nu) \text{ odd}} \theta_{\mu, \nu}(u, Z), \quad u = (u_1, u_2)$$

about the origin, where for $\mu, \nu \in \{0, 1\}$

$$\theta_{\mu, \nu}(u, Z) := \sum_{m \in \mathbb{Z}^2 + \frac{1}{2}\mu} e^{\pi i m^t Z m} e^{2\pi i m^t (u + \frac{1}{2}\nu)}, \quad u = (u_1, u_2) \in \mathbb{C}^2, \quad Z \in \mathcal{H}_2$$

is the theta function with characteristics (see [Mu]). Then the canonically polarized Jacobian of the hyperelliptic curve over \mathbb{C} determined by the equation

$$y^2 = f_Z(x, 1)$$

corresponds to Z . (There are six theta functions with odd characteristics and hence f_Z is a sextic, i.e. homogeneous of degree 6.)

The difficulty in applying this formula is to know how to normalize the sextic f_Z properly to guarantee that its coefficients are algebraic integers as well as finding similar expressions for its Galois conjugates. In general, this would be accomplished by an application of Shimura's general reciprocity law. We would obtain rapidly convergent series giving the minimal polynomials of these coefficients. Since the coefficients of the minimal polynomials of the coefficients of the sextic are in \mathbb{Z} , truncating the series would then allow us to compute them exactly. We show how this works for our special case in §4.

3 Principal Polarizations

From now on we assume that \mathcal{O} is the ring of integers of $K = \mathbb{Q}(\sqrt{-N}) \subset \mathbb{C}$, $N \equiv 3 \pmod{4}$ prime and the class number of \mathcal{O} is 1. Hence, $E_1 = E_2 = E$, with E isomorphic to \mathbb{C}/\mathcal{O} over \mathbb{C} .

The principal polarizations of $E \times E$ up to isomorphism correspond to positive definite unimodular Hermitian forms of rank 2 over \mathcal{O} up to $GL_2(\mathcal{O})$ -equivalence. In order to find a set of representatives of these Hermitian forms we will exploit the happy accident that since we assume $N \equiv 3 \pmod{4}$ the quaternion algebra $B = (\frac{-1, -N}{\mathbb{Q}})$ (up to isomorphism the unique quaternion algebra over \mathbb{Q} ramified only at N and ∞) contains $\mathbb{Q}(i)$. This allows us to convert the question to that of finding Hermitian forms over $\mathbb{Z}[i]$ of discriminant $-N$ up to equivalence and this is quite simple. Here is how it works.

Consider in B the order

$$R = \mathbb{Z} + \mathbb{Z}i + \mathbb{Z}\frac{1}{2}(1+j) + \mathbb{Z}i\frac{1}{2}(1+j), \quad i^2 = -1, j^2 = -N.$$

R is a maximal order in B with a natural embedding of \mathcal{O} sending $\sqrt{-N}$ to j . The rank 2 unimodular Hermitian forms arising from polarizations of $E \times E$ correspond to rank 1 left R -modules.

Since R also has an embedding of $\mathbb{Z}[i]$ (sending i to i) we may associate to a left R -module a rank 2 Hermitian form Φ over $\mathbb{Z}[i]$. We can give Φ as a triple (a, b, c) with $a, c \in \mathbb{Z}_{>0}$ and $b \in \mathbb{Z}[i]$, where $\Phi(u, v) = 2au\bar{u} + bu\bar{v} + \bar{b}\bar{u}v + 2cv\bar{v}$. It is not hard to see that this form has discriminant $b\bar{b} - 4ac = -N$.

It will be more convenient to work with SL_2 rather than GL_2 equivalence and to avoid duplications we consider only forms $\Phi = (a, b, c)$ with $b \equiv 1 \pmod{2}$. The above discussion establishes a 1-1 correspondence between principal polarizations on $E \times E$, up to $SL_2(\mathcal{O})$ -equivalence, and positive definite binary Hermitian forms $\Phi = (a, b, c)$ over $\mathbb{Z}[i]$ of discriminant $-N$, up to $SL_2(\mathbb{Z}[i])$ -equivalence.

Let \mathbb{H} be the hyperbolic 3-space

$$\mathbb{H} = \{w = (x, y, t) \in \mathbb{R}^3 \mid t > 0\},$$

which we will think as embedded in the Hamilton quaternion algebra H by $(x, y, t) \mapsto x + iy + jt$ (here i, j are the usual basis of H with $i^2 = j^2 = -1$ and $ij = -ji$). If $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{C})$ then it is not hard to check that

$$w \mapsto (aw + b)(cw + d)^{-1}$$

sends \mathbb{H} to \mathbb{H} defining an action of $SL_2(\mathbb{C})$ on \mathbb{H} and in particular, an action of $SL_2(\mathbb{Z}[i])$. This last action has a very simple fundamental domain, whose closure is given by $w = (x, y, t) \in \mathbb{H}$ with $x^2 + y^2 + t^2 \geq 1, x \leq 1/2, y \leq 1/2, 0 \leq x + y$.

We can associate to a form Φ the point $w = (b + \sqrt{N}j)/2a \in \mathbb{H}$ and we call Φ *reduced* if w lies in the fundamental domain. The action of $SL_2(\mathbb{Z}[i])$ on \mathbb{H} mimics that on Hermitian forms. Every form is $SL_2(\mathbb{Z}[i])$ -equivalent to a unique reduced form.

The situation is in fact very analogous to that of positive definite binary quadratic forms over \mathbb{Z} and, as in that case, it is easy to write an algorithm that lists all reduced forms Φ of a given discriminant (we do not really need N to be prime or class number 1 for this). Here is a brief sketch (all of this is classical going back to Hermite [He, I, p. 251]).

Input: $N \equiv 3 \pmod{4}$

```

For  $0 \leq r, s \leq \sqrt{N/2}$ ,  $r$  odd,  $s$  even
  Set  $m := (r^2 + s^2 + N)/4$ 
  For  $a|m$ ,  $\max(r, s) \leq a \leq \sqrt{m}$ 
    Add  $(a, r + is, m/a)$  to List
    Add  $(a, -r + si, m/a)$  to List unless
       $a = m/a$  or  $r = a$  or  $s = a$  or  $s = 0$ 

```

Output: List

As an example, we give in table 1 the list of reduced forms of discriminant -163 .

Table 1. Reduced Hermitian forms over $\mathbb{Z}[i]$ of discriminant -163

(1, 1, 41)
(2, 1 + 2i, 21)
(3, ±1 + 2i, 14)
(6, ±1 + 2i, 7)
(4, ±3 + 2i, 11)
(6, ±5 + 2i, 8)
(5, ±1 + 4i, 9)
(7, ±5 + 6i, 8)

In general, the number of Φ 's is the *class number* n of B [Ei], which can be given in terms of N as follows (a formula valid for any prime $N \equiv 3 \pmod{4}$)

$$n = \begin{cases} \frac{1}{12}(N+5), & \left(\frac{-3}{N}\right) = +1 \\ \frac{1}{12}(N+13), & \left(\frac{-3}{N}\right) = -1. \end{cases}$$

Finally, given a $\Phi = (a, b, c)$ as above, $b = r + si$, the matrix

$$Z_\Phi := \frac{1}{2a} \begin{pmatrix} r + \sqrt{-N} & s \\ s & -r + \sqrt{-N} \end{pmatrix} \in \mathcal{H}_2$$

is a period matrix corresponding to the associated principal polarization on $E \times E$.

4 Bolza-Klein Sextics

The product polarization on $E \times E$ corresponds to the reduced form $\Phi = (1, 1, (N+1)/4)$ in the principal class; hence, forms Φ not in the principal class correspond to curves.

Given a form $\Phi = (a, b, c)$ not in the principal class we define the associated normalized Bolza–Klein sextic f_Φ as follows.

$$f_\Phi(u_1, u_2) := \frac{1}{a^6 |\eta((1 + \sqrt{-N})/2)|^{24}} f_{Z_\Phi}(u_1, u_2),$$

where η is Dedekind’s eta function. It satisfies the following properties.

- The $SL_2(\mathbb{C})$ class of f_Φ depends only on the $SL_2(\mathbb{Z}[i])$ class of Φ .
- f_Φ has coefficients in K and $a^6 f_\Phi$ has coefficients in \mathcal{O} .
- The Igusa invariants [Ig] of f_Φ are in \mathbb{Z} and depend only on the $SL_2(\mathbb{Z}[i])$ -equivalence class of Φ .

The genus 2 curve

$$C_\Phi : \quad y^2 = f_\Phi(x, 1)$$

is then defined over K and, over the algebraic closure \overline{K} of K in \mathbb{C} , its Jacobian is isomorphic to $E \times E$.

Given a form $\Phi = (a, b, c)$ let $\Phi^\iota := (a, -\bar{b}, c)$. Suppose both Φ and Φ^ι are reduced. Then Φ and Φ^ι are not $SL_2(\mathbb{Z}[i])$ -equivalent but they are (always) $GL_2(\mathbb{Z}[i])$ -equivalent. The corresponding curves C_Φ, C_{Φ^ι} are hence isomorphic over \overline{K} ; note that they are also complex conjugates of each other. Otherwise, curves C_Φ corresponding to different reduced forms are non-isomorphic. The involution ι has a natural counterpart on the left R -ideals in B and it turns out that the number of orbits of ι is what is classically known as the *type number* of B [Ei]. Hence, there are $t - 1$ isomorphism classes of curves with Jacobian isomorphic to $E \times E$, where t is the type number of the quaternion algebra B [HN].

Here is a table with the values of n and t for the primes N we are considering.

Table 2. Type and class number of the quaternion algebra B

N	n	t
3	1	1
7	1	1
11	2	2
19	2	2
43	4	3
67	6	4
163	14	8

Note that for $N = 3$ or 7 we only have the product polarization and hence there is no curve C with unpolarized Jacobian isomorphic to $E \times E$ in that case.

Given a curve C defined over $\overline{\mathbb{Q}}$ its *field of moduli* is the field $F \subset \overline{\mathbb{Q}}$ characterized by the property: For every $\tau \in Gal(\overline{\mathbb{Q}}/\mathbb{Q})$, C^τ is isomorphic to C if and

only if τ is the identity on F . Clearly isomorphic curves have the same field of moduli. Notice that F is the smallest field over which a curve isomorphic to C *could* be defined, but it is not in general a field over which it *can* be defined. In fact, for example, Shimura showed that no generic hyperelliptic curve of even genus has a model over its field of moduli [Sh Thm 3]. See [Me] for a discussion of this issue for curves of genus 2.

For the curves C_Φ the field of moduli is \mathbb{Q} (the field generated by the Igusa invariants [Ig]), but, in fact, most are not definable over \mathbb{Q} ; only those forms Φ which are $SL_2(\mathbb{Z}[i])$ -equivalent to Φ^ι give rise to curves definable over \mathbb{Q} .

To see this we note that by their very construction the period matrices Z_Φ lies in a certain real 3-dimensional cycle in \mathcal{H}_2 considered by Shimura [Sh]. Namely, the cycle defined by

$$Z \in \mathcal{H}_2, \quad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} Z = -\overline{Z} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

If A_Z is the complex abelian surface corresponding to such a Z then there is an isomorphism

$$\lambda : A_Z \longrightarrow \overline{A_Z}, \quad \text{with} \quad \overline{\lambda} \circ \lambda = -\text{id}.$$

(Applied to Z_Φ this yields the fact that the curves C_Φ and C_{Φ^ι} are both isomorphic and complex conjugate to each other as mentioned above).

It follows that if A_Z has no automorphisms other than $\pm\text{id}$ then it has no model defined over its field of moduli. It is not hard to see that this holds for Z_Φ , for every Φ in the interior of the fundamental domain.

5 Examples

We end with an illustration of the above discussion, giving the outcome of algorithm when $N = 43$. The calculations were done using PARI-GP. The routines as well as the data for all cases is available at:

<http://www.ma.utexas.edu/users/villegas>

The reduced forms Φ of discriminant -43 are $(1, 1, 11)$, $(2, 1 + 2i, 6)$ and $(3, \pm 1 + 2i, 4)$.

1) For $\Phi = (2, 1 + 2i, 6)$ we obtain

$$f_\Phi(x, 1) = \frac{1}{2}(-x^6 + \frac{1}{2}(-3 + 567\sqrt{-43})x^4 + \frac{1}{2}(3 + 567\sqrt{-43})x^2 + 1)$$

Its Igusa invariants are

$$\begin{aligned} J_2 &= 1728012 \\ J_4 &= 93313728006 \\ J_6 &= -186622271996 \\ J_8 &= -2176943579975806271997 \\ J_{10} &= 2176782336000000000000000 \end{aligned}$$

(these were calculated using classical algorithms for invariants of a sextic following Mestre [Me]).

As in the example of the introduction $D = J_{10} = 2^{-12} \operatorname{disc}(f_\Phi)$ factors nicely

$$D = (2^2 \cdot 3 \cdot 5)^{12}.$$

This curve descends to \mathbb{Q} ; here is a model

$$y^2 = x^6 + 24384x^5 + 61311x^4 + 585856x^3 + 813483x^2 + 3214656x + 1472877.$$

2) For $\Phi = (3, 1+2i, 4)$ we obtain

$$\begin{aligned} f_\Phi(x, 1) = \frac{4}{3^3} & ((14\sqrt{-43} - 160)x^6 + (42\sqrt{-43} + 162)x^5 + \\ & (2247\sqrt{-43} - 159)x^4 + 17021x^3 + \\ & (2247\sqrt{-43} + 159)x^2 + (-42\sqrt{-43} + 162)x \\ & + 14\sqrt{-43} + 160) \end{aligned}$$

Its Igusa invariants are

$$\begin{aligned} J_2 &= 14333772 \\ J_4 &= 7393823156166 \\ J_6 &= 3726840435157546564 \\ J_8 &= -312234946681873274015037 \\ J_{10} &= 73558275113866410000000000000000 \end{aligned}$$

and

$$D = J_{10} = (2 \cdot 3 \cdot 5 \cdot 7)^{12}.$$

As explained in §4, since Φ corresponds to a point in the interior of the fundamental domain the curve C_Φ has no model over \mathbb{Q} . Alternatively, we can see this following Mestre [Me]. When the curve has no extra automorphisms (i.e. its only automorphisms are the identity and the hyperelliptic involution), the obstruction to the curve being definable over its field of moduli (\mathbb{Q} in our case) is given by a conic in \mathbb{P}^2

$$xMx^t = 0, \quad x = (x_1 : x_2 : x_3) \in \mathbb{P}^2,$$

where M is a 3×3 symmetric matrix whose entries are certain invariants of the sextic; more precisely, the curve is definable over its field of moduli if and only if the conic has a rational point there. Explicitly, we have $M = (m_{i,j})$ with (we have actually simplified slightly the matrix given by Mestre)

$$\begin{aligned} m_{11} &= 3J_2^3 - 160J_4J_2 - 3600J_6 \\ m_{21} &= -J_4J_2^2 + 330J_6J_2 + 160J_4^2 \\ m_{31} &= -J_6J_2^2 - 840J_6J_4 - 8000J_{10} \\ m_{22} &= -25J_6J_2^2 - 8J_4^2J_2 - 120J_6J_4 - 2000J_{10}, \\ m_{32} &= 67J_6J_4 + 600J_{10}J_2 + 90J_6^2 \\ m_{33} &= -33J_6^2J_2 - 100J_6J_4^2 - 800J_{10}J_4 \end{aligned}$$

where $J_2, J_4, J_6, J_8, J_{10}$ are the Igusa invariants.

In our case we have

$$\begin{aligned} m_{11} &= -21538723388574481387776 \\ m_{12} &= 24856361223852137345176064256 \\ m_{13} &= -23971255400369899892885589544571136 \\ m_{22} &= -28732882146400381994651008552571136 \\ m_{23} &= 27776672840855638207256856144392139100416 \\ m_{33} &= -26987491534155851141341724256178812956900004096 \end{aligned}$$

We easily verify that this conic has rational points everywhere locally except at the primes 43 and ∞ ; in particular, it has no rational points.

We should point out that the vanishing of the determinant of M precisely corresponds to the curve having extra automorphisms. As with D , this determinant factors nicely

$$\det M = -2^{64} \cdot 3^{38} \cdot 5^{34} \cdot 7^{28} \cdot 19^4 \cdot 29^2 \cdot 37^2 \cdot 43 .$$

Finally, let p be a prime which splits in $K = \mathbb{Q}(\sqrt{-43})$ as $p = \mathcal{P}\bar{\mathcal{P}}$. The reduction of the curve C_ϕ modulo \mathcal{P} gives a smooth curve \overline{C} of genus 2 over \mathbb{F}_p . We have verified that for all primes in the range $167 \leq p < 10000$ such that $4p = a^2 + 43$ for some $a \in \mathbb{N}$ the curve \overline{C} or its quadratic twist attains the maximum number of points possible, namely $p+1+2[\sqrt{2p}]$ (an improvement on Weil bounds due to Serre).

References

- [Ei] Eichler, M. Zur Zahlentheorie der Quaternion-Algebren. *J. Reine Angew. Math.* **195** (1955) 127–151
- [GZ] Gross, B. and Zagier, D. On singular moduli. *J. Reine Angew. Math.* **355** (1985) 191–220
- [HN] Hayashida, T. and Nishi, M. On certain type of Jacobian varieties of dimension 2. *Natur. Sci. Rep. Ochanomizu Univ.* **16** (1965) 49–57
- [He] Hermite, Ch.: *Oeuvres de Charles Hermite*, Gauthiers–Villars, Paris, 1905
- [Ig] Igusa, J.: Arithmetic variety of moduli for genus two. *Ann. of Math.* **72** (1960) 612–649
- [Me] Mestre, J.-F.: Construction de courbes de genre 2 à partir de leurs modules. Effective methods in algebraic geometry (Castiglioncello, 1990) 313–334, *Progr. Math.* **94**, Birkhäuser, 1991.
- [Mu] Mumford, D.: *Tata lectures on Theta I*. Birkhäuser, 1983.
- [NN] Narasimhan, M. S. and Nori, M. V.: Polarisations on an abelian variety. *Proc. Indian Acad. Sci. Math. Sci.* **90** (1981) 125–128
- [Sh] Shimura, G.: On the field of rationality for an abelian variety. *Nagoya Math. J.* **45** (1972) 167–178
- [vW] van Wamelen, P.: Examples of genus two CM curves defined over the rationals. *Math. Comp.* **68** (1999) 307–320

Reduction in Purely Cubic Function Fields of Unit Rank One

Renate Scheidler

Department of Mathematical Sciences
University of Delaware, Newark DE 19716, USA
`scheidle@math.udel.edu`

Abstract. This paper analyzes reduction of fractional ideals in a purely cubic function field of unit rank one. The algorithm is used for generating all the reduced principal fractional ideals in the field, thereby finding the fundamental unit or the regulator, as well as computing a reduced fractional ideal equivalent to a given nonreduced one. It is known how many reduction steps are required to achieve either of these tasks, but not how much time and storage each reduction step takes. Here, we investigate the complexity of a reduction step, the precision required in the approximation of the infinite power series that occur throughout the algorithm, and the size of the quantities involved.

1 Introduction and Motivation

Basis reduction of fractional ideals is one of the key ingredients in the computation of invariants of a purely cubic function field of unit rank one, such as the fundamental unit, the regulator, the ideal class number and, most importantly, the order of the Jacobian of the field. In fields of characteristic at least five, a basis reduction procedure was first presented in [2], and its discussion was continued in [1]. The algorithm was originally used for generating the entirety of reduced fractional principal ideals and thus finding the fundamental unit and the regulator of the field. Unfortunately, there are usually exponentially many such ideals, and enumerating them all is not the most efficient method for computing the regulator. This is where another aspect of ideal basis reduction comes into play: it quickly produces from a given nonreduced fractional ideal an equivalent reduced one.

The infrastructure of the set of reduced principal ideals is a powerful tool for invariant computations and a variety of other applications in both computational number theory and cryptography. Loosely speaking, the product of two reduced fractional principal ideals is generally not reduced; however, reduction produces a reduced ideal “close to” the product ideal, and the number of basis reduction steps required is polynomial in the size of the field. This phenomenon can be exploited for computing invariants of the field much faster than with the naive approach outlined above. For hyperelliptic, i.e. quadratic function fields (where reduction amounts to computing a simple continued fraction expansion), this

was successfully accomplished in [3] with an improvement in complexity from p to essentially $p^{2/5}$, where p is the number of reduced fractional principal ideals. Work on the purely cubic setting is in progress at the time of writing, and we expect a similarly dramatic speed-up from our original method of [2].

While it is known how many reduction steps are required to compute the fundamental unit and the regulator of a purely cubic function field of unit rank one and characteristic at least five, it is as yet unclear how long an individual reduction step takes, how large the inputs and outputs get, and how much “precision” is required. Numerical experiments and heuristics suggest that the answers to these three questions are ‘not very long’, ‘not very large’, and ‘not too much’, respectively — at least in the reduced case — but we lack proof. This paper remedies this rather unsatisfactory situation. To that extent, we provide answers to the following questions:

- What is the complexity of an ideal basis reduction step?
- What is the size of the quantities involved?
- What is the minimal precision required in the approximation of the infinite series involved?

2 Purely Cubic Function Fields

A detailed treatment of this material can be found in [2] and [1]. A *purely cubic function field* is the function field of a plane curve given by the (not necessarily nonsingular) model $y^3 - D(x) = 0$ over a finite field $k = \mathbb{F}_q$ of order q whose characteristic is not 3; here, $D(x) \in k[x]$ is a cubefree polynomial. Thus, a purely cubic function field can be viewed as a cubic extension $K = k(x)(\rho)$ of a rational function field $k(x)$ obtained by adjoining a cube root ρ of a cubefree polynomial $D = D(x) \in k[x]$; this makes it the function field analogue of a purely cubic number field. We write $D = GH^2$ where $G, H \in k[x]$ are squarefree and coprime and $\deg(G) \geq \deg(H)$.

The integral closure \mathcal{O} of $k[x]$ in K is both a ring and a $k[x]$ -module of rank 3 that is generated by the *integral basis* $\{1, \rho, \omega\}$ where $\omega = \rho^2/H$, so ω is a cube root of $\bar{D} = G^2H$. If $\alpha = a + b\rho + c\omega \in K$ ($a, b, c \in k(x)$), then the *conjugates* of α are $\alpha' = a + b\rho + c\omega^2$ and $\alpha'' = a + b\omega\rho + c\omega$ where ι is a fixed primitive cube root of unity. The *norm* of α is $N(\alpha) = \alpha\alpha'\alpha'' = a^3 + b^3GH^2 + c^3G^2H - 3abcGH \in k(x)$.

We henceforth make the following assumptions:

- $q \equiv -1 \pmod{3}$ (so k contains no primitive cube roots of unity),
- $\deg(D) \equiv 0 \pmod{3}$,
- The leading coefficient $\text{sgn}(D)$ of D is a cube in $k^* = k \setminus \{0\}$.

Then $K/k(x)$ has two points at infinity, namely one rational point and one quadratic point. The former gives rise to an embedding of K into the field $k\langle x^{-1} \rangle$ of *Puiseux series* over k , and the Galois closure of K is embedable into $k(\iota)\langle x^{-1} \rangle$; nonzero elements in $k\langle x^{-1} \rangle$ (respectively, $k(\iota)\langle x^{-1} \rangle$) have the form $\alpha = \sum_{i=-m}^{\infty} a_i x^{-i} = \sum_{i=-\infty}^m a_{-i} x^i$ with $a_i \in k$ (respectively, $k(\iota)$) for $i \geq -m$ and $a_{-m} \neq 0$. The degree valuation on $k(x)$ extends canonically to $k\langle x^{-1} \rangle$ via

$\deg(\alpha) = m$ and to $k(\iota)\langle x^{-1} \rangle$ via $\deg(\alpha + \beta\iota) = (\deg(\alpha + \beta\iota)(\alpha + \beta\iota^2)/2 = \deg(\alpha^2 - \alpha\beta + \beta^2)/2 \in \mathbb{Z}$ ($\alpha, \beta \in k\langle x^{-1} \rangle$). For $\alpha = \sum_{i=-m}^{\infty} a_i x^{-i} \in k\langle x^{-1} \rangle$, we set $|\alpha| = q^{\deg(\alpha)}$, $\text{sgn}(\alpha) = a_{-m}$, and $[\alpha] = \sum_{i=0}^m a_{-i} x^i$ (with $|0| = 0$ and $[0] = 0$). For $\alpha \in K$, we have $|\alpha'| = q^{\deg(\alpha')} = |\alpha'\alpha''|^{1/2}$. Note that $|G| \geq |H|$ implies $|\rho| \leq |\omega|$.

Under the above assumptions, K has *unit rank* 1 over $k(x)$; that is, the group \mathcal{O}^* of units in \mathcal{O} is isomorphic to $k^* \times \mathbb{Z}$ (see Theorem 2.1 of [2]). A generator ϵ of the torsionfree part of \mathcal{O}^* is a *fundamental unit* of $K/k(x)$. If ϵ has positive degree (and is hence unique up to constant factors), then $R = \deg(\epsilon)/2 = -\deg(\epsilon')$ is the *regulator* of $K/k(x)$.

3 Fractional Ideals

A *fractional ideal* (of \mathcal{O}) is a subset \mathfrak{f} of K such that there exists a nonzero $d \in k[x]$ such that $d\mathfrak{f}$ is an integral ideal in \mathcal{O} , i.e. an additive subgroup of \mathcal{O} that is also closed under multiplication by elements of \mathcal{O} . The unique monic polynomial $d = d(\mathfrak{f})$ of minimal degree that satisfies this condition is the *denominator* of \mathfrak{f} . \mathfrak{f} is *principal* if it consists of \mathcal{O} -multiples of some $\theta \in K$; write $\mathfrak{f} = (\theta)$. The fractional ideals form an infinite Abelian group \mathcal{I} under multiplication, of which the set of principal fractional ideals forms an infinite subgroup \mathcal{P} . The factor group \mathcal{I}/\mathcal{P} is the *ideal class group* of $K/k(x)$; it is a finite Abelian group whose order is the *ideal class number* of $K/k(x)$. The product $h = Rh'$ where R is the regulator of $K/k(x)$ is the order of the group of k -rational points on the *Jacobian* of K ; it is independent of the element x and thus the representation of K as a function field. Two fractional ideals are *equivalent* if lie in the same coset in \mathcal{I}/\mathcal{P} , i.e. if they differ by a factor that is a principal fractional ideal.

We will henceforth assume “fractional ideal” to mean “nonzero fractional ideal containing 1”. Then every fractional ideal \mathfrak{f} is a $k[x]$ -module of rank 3 with a basis $\{1, \mu, \nu\}$; write $\mathfrak{f} = [1, \mu, \nu]$. If $\mathfrak{f} = [1, \mu, \nu]$ where $\mu = (m_0 + m_1\rho + m_2\omega)/d$, $\nu = (n_0 + n_1\rho + n_2\omega)/d$ with $m_0, m_1, m_2, n_0, n_1, n_2, d \in k[x]$ jointly coprime and $d = d(\mathfrak{f})$, then the *norm* of \mathfrak{f} is $N(\mathfrak{f}) = a(m_1n_2 - m_2n_1)/d^2 \in k(x)$ where $a \in k^*$ is chosen so that $N(\mathfrak{f})$ is monic. The *discriminant* of \mathfrak{f} is

$$\Delta(\mathfrak{f}) = \det \begin{pmatrix} 1 & 1 & 1 \\ \mu & \mu' & \mu'' \\ \nu & \nu' & \nu'' \end{pmatrix}^2 \in k(x).$$

Both $N(\mathfrak{f})$ and $\Delta(\mathfrak{f})$ (up to a constant factor) are independent of the choice of $k[x]$ -basis of \mathfrak{f} , and $N(\mathfrak{f})$ is multiplicative on the set of fractional ideals.

A *canonical basis* of a fractional ideal \mathfrak{f} is a $k[x]$ -basis $\{1, \alpha, \beta\}$ where $\alpha = s'(u + \rho)/s$, $\beta = s''(v + w\rho + \omega)/s$ with $s, s', s'', u, v, w \in k[x]$, $s's''$ divides s , s'' divides H , and $\gcd(s', H) = 1$. Here $s = d(\mathfrak{f})$ up to sign, and we may assume $|s'u|, |s''v| < |s|$, and $|w| < |s''|$. Such a basis always exists, and it is a simple matter to generate a canonical basis from any given basis, or compute a canonical basis of the product ideal of two fractional ideals given in terms of respective canonical bases (see [1]).

An element θ in a fractional ideal \mathfrak{f} is a *minimum* in \mathfrak{f} if for any $\phi \in \mathfrak{f}$, $|\phi| \leq |\theta|$ and $|\phi'| \leq |\theta'|$ imply $\phi \in k\theta$; that is, ϕ differs from θ only by a constant factor. \mathfrak{f} is *reduced* if 1 is a minimum in \mathfrak{f} . It is easy to see that an element θ is a minimum in \mathcal{O} if and only if the fractional principal ideal $\mathfrak{f} = (\theta^{-1})$ is reduced.

We summarize some properties of fractional ideals; the proofs of these results can be found in [2] and [1].

Proposition 3.1. *Let \mathfrak{f} be a fractional ideal.*

1. $\Delta(\mathfrak{f}) = a^2 N(\mathfrak{f})^2 \Delta$ for some $a \in k^*$.
2. $|d(\mathfrak{f})|^{-2} \leq |N(\mathfrak{f})| \leq |d(\mathfrak{f})|^{-1}$.
3. If \mathfrak{f} is reduced, then $|\Delta(\mathfrak{f})| > 1$, so $|N(\mathfrak{f})| > |\Delta|^{-1/2}$.
4. If \mathfrak{f} is reduced, then $|d(\mathfrak{f})| < |\Delta|^{1/2}$, so $|N(\mathfrak{f})| < |\Delta||d(\mathfrak{f})|^{-3}$.
5. If $|\Delta(\mathfrak{f})| > |d(\mathfrak{f})|^2$, i.e. $|d(\mathfrak{f})| < |N(\mathfrak{f})||\Delta|^{1/2}$, then \mathfrak{f} is reduced.
6. If \mathfrak{f} is nonreduced, then $|N(\mathfrak{f})| \leq |\Delta|^{-1/4}$, so $|\Delta(\mathfrak{f})| \leq |\Delta|^{1/2}$.

Let \mathfrak{f} be a fractional ideal and let θ be a minimum in \mathfrak{f} . An element $\phi \in \mathfrak{f}$ is the *neighbor* of θ in \mathfrak{f} if ϕ is also a minimum in \mathfrak{f} , $|\theta| < |\phi|$, and for no $\psi \in \mathfrak{f}$, $|\theta| < |\psi| < |\phi|$ and $|\psi'| < |\theta'|$. ϕ always exists and is unique up to nonzero constant factors (see Theorem 5.1 of [2]).

The *Voronoi chain* $(\theta_n)_{n \in \mathbb{N}}$ of successive minima in \mathcal{O} where $\theta_1 = 1$ and θ_{n+1} is the neighbor of θ_n in \mathcal{O} yields the entirety of minima in \mathcal{O} of nonnegative degree (Voronoi first investigated this chain in cubic number fields in [4]). This chain is given by the recurrence $\theta_{n+1} = \mu_n \theta_n$ where μ_n is the neighbor of 1 in the reduced fractional principal ideal $\mathfrak{f}_n = (\theta_n^{-1})$ ($n \in \mathbb{N}$). The first nontrivial unit $\epsilon = \theta_{p+1}$ ($p \in \mathbb{N}$) encountered in this chain is the fundamental unit of K of nonnegative degree. Since the recurrence for the Voronoi chain implies $\theta_{mp+n} = \epsilon^m \theta_n$ for $m \in \mathbb{N}_0$ and $n \in \mathbb{N}$, $\{\mathfrak{f}_1, \mathfrak{f}_2, \dots, \mathfrak{f}_p\}$ is the complete set of reduced principal fractional ideals in K . The positive integer p is the *period* of ϵ . By Theorem 6.5 of [2], $p = O(q^{(\deg(\Delta)/2)-2})$, so there may be (and usually are) exponentially many reduced fractional ideals in $K/k(x)$.

4 Reduced Bases

For the remainder of the paper, we exclude the case of even characteristic, so k has characteristic at least 5. For $\theta = l + m\rho + n\omega \in K$ with $l, m, n \in k(x)$, we define

$$\begin{aligned} \xi_\theta &= \theta - l &= m\rho + n\omega, \\ \eta_\theta &= (1 + 2\iota)^{-1}(\theta' - \theta'') = m\rho - n\omega, \\ \zeta_\theta &= \theta' + \theta'' &= 2l - m\rho - n\omega, \end{aligned} \tag{4.1}$$

where $\iota (\not\in k)$ is a primitive cube root of unity. Then

$$\theta = \frac{1}{2}(3\xi_\theta + \zeta_\theta), \quad \theta'\theta'' = \frac{1}{4}(3\eta_\theta^2 + \zeta_\theta^2), \tag{4.2}$$

so

$$|\theta'| = \max\{|\eta_\theta|, |\zeta_\theta|\}, \quad |\xi_\theta| \leq \max\{|\theta|, |\theta'|\}. \tag{4.3}$$

If $\{1, \theta, \phi\}$ is a basis of a fractional ideal \mathfrak{f} , then

$$(\xi_\mu \eta_\nu - \xi_\nu \eta_\mu)^2 = -\frac{4}{27} \Delta(\mathfrak{f}). \quad (4.4)$$

A $k[x]$ -basis $\{1, \mu, \nu\}$ of a (reduced or nonreduced) fractional ideal \mathfrak{f} is *reduced* if

$$\begin{aligned} |\xi_\mu| &> |\xi_\nu|, \quad |\zeta_\mu| < 1, \quad |\zeta_\nu| \leq 1, \quad |\eta_\mu| < 1 \leq |\eta_\nu|, \\ \text{if } |\eta_\nu| &= 1, \text{ then } |\nu| \neq 1. \end{aligned} \quad (4.5)$$

The following procedure (which is essentially Algorithm 7.1 in [2]) generates a reduced basis of a fractional ideal.

Algorithm 4.1. (Ideal Basis Reduction)

Input: $\tilde{\mu}, \tilde{\nu}$ where $\{1, \tilde{\mu}, \tilde{\nu}\}$ is a basis of some fractional ideal \mathfrak{f} .

Output: μ, ν where $\{1, \mu, \nu\}$ is a reduced basis of \mathfrak{f} .

Algorithm:

1. Set $\mu = \tilde{\mu}, \nu = \tilde{\nu}$.

2. If $|\xi_\mu| < |\xi_\nu|$ or if $|\xi_\mu| = |\xi_\nu|$ and $|\eta_\mu| < |\eta_\nu|$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

3. If $|\eta_\mu| \geq |\eta_\nu|$

3.1. While $|\xi_\nu \eta_\nu| > |\Delta(\mathfrak{f})|^{1/2}$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_\mu / \xi_\nu \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

3.2. Replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_\mu / \xi_\nu \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

3.3. If $|\eta_\mu| = |\eta_\nu|$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 1 & -\text{sgn}(\eta_\mu \eta_\nu^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

4. While $|\eta_\nu| < 1$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \eta_\nu / \eta_\mu \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

While $|\eta_\mu| \geq 1$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} \lfloor \eta_\nu / \eta_\mu \rfloor & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

5. Replace μ by $\mu - \lfloor \zeta_\mu \rfloor / 2$ and ν by $\nu - \lfloor \zeta_\nu \rfloor / 2$.
6. If $|\xi_\nu| = |\eta_\nu| = 1$, replace ν by $\nu - \lfloor \nu \rfloor$.

A reduced basis provides an easy means by which to determine whether or not an ideal is reduced (see [1]):

Proposition 4.2. *Let $\{1, \mu, \nu\}$ be a reduced basis of a fractional ideal \mathfrak{f} .*

1. *If \mathfrak{f} is reduced, then μ is the neighbor of 1 in \mathfrak{f} .*
2. *\mathfrak{f} is reduced if and only if $|\mu| > 1$ and $\max\{|\nu|, |\eta_\nu|\} > 1$.*
3. *\mathfrak{f} is nonreduced if and only if $|\mu| \leq 1$ or $|\nu| < |\eta_\nu| = 1$.*

Part 2 of Proposition 4.2 in conjunction with (4.3) and step 5 of Algorithm 4.1 implies that step 6 can only be entered if the input ideal \mathfrak{f} is nonreduced. Part 1 of this proposition together with the recursion for the Voronoi chain shows that repeated application of Algorithm 4.1 to the ideal $\mathfrak{f}_n = (\theta_n^{-1})$ with subsequent division of \mathfrak{f}_n by the neighbor μ_n of 1 in \mathfrak{f}_n generates all the minima of nonnegative degree in \mathcal{O} and hence the fundamental unit of K . A similar recursion allows for computing from a given nonreduced fractional ideal an equivalent reduced one.

Let \mathfrak{f} be any nonreduced fractional ideal and define a sequence $(\mathfrak{f}_n)_{n \in \mathbb{N}}$ of pairwise equivalent fractional ideals as follows.

$$\mathfrak{f}_1 = \mathfrak{f}, \quad \mathfrak{f}_{n+1} = (\phi_n^{-1})\mathfrak{f}_n \quad \text{where} \quad \phi_n = \begin{cases} \mu_n & \text{if } |\mu_n| \leq 1, \\ \nu_n & \text{if } |\mu_n| > 1, \end{cases} \quad (n \in \mathbb{N}) \quad (4.6)$$

and $\{1, \mu_n, \nu_n\}$ is a reduced basis of \mathfrak{f}_n . The case $\phi_n = \nu_n$ in (4.6) can happen at most once; that is, if \mathfrak{f}_n is nonreduced with $|\mu_n| > 1$, then \mathfrak{f}_{n+1} is reduced and a reduced basis of \mathfrak{f}_{n+1} can be obtained directly without applying Algorithm 4.1:

Proposition 4.3. *Let \mathfrak{f} be a nonreduced ideal with a reduced basis $\{1, \mu, \nu\}$ and let $\mathfrak{g} = (\nu^{-1})\mathfrak{f} = [1, \mu\nu^{-1}, \nu^{-1}]$. If $|\mu| > 1$, then \mathfrak{g} is reduced with a reduced basis $\{1, \mu\nu^{-1}, \nu^{-1}\}$.*

Proof. If $|\mu| > 1$, then by part 3 of Proposition 4.2 and (4.3) $|\nu| < 1 = |\eta_\nu| = |\nu'|$. Let $\alpha = \mu\nu^{-1}$ and $\beta = \nu^{-1}$. Then $|\alpha'| = |\mu'| < 1$, so $|\eta_\alpha| < 1$, $|\zeta_\alpha| < 1$, and since $|\alpha| > 1$, $|\xi_\alpha| = |\alpha|$ by (4.2). Furthermore, $|\beta'| = 1$, so $|\zeta_\beta| \leq 1$, and $|\xi_\beta| = |\beta| = |\nu|^{-1} > 1$. Since $\eta_\beta = -\eta_\nu(\nu'\nu'')^{-1}$, $|\eta_\beta| = 1$.

Since $|\alpha| > 1$ and $\max\{|\beta|, |\eta_\beta|\} > 1$, \mathfrak{g} is reduced by part 2 of Proposition 4.2. Since $|\xi_\alpha| = |\alpha| > |\nu|^{-1} = |\xi_\beta|$, $|\eta_\alpha| < 1 = |\eta_\beta|$, $|\zeta_\alpha| < 1$, and $|\zeta_\beta| \leq 1$, $\{1, \alpha, \beta\}$ is a reduced basis of \mathfrak{g} .

A polynomial number of steps of recursion (4.6) produces a reduced ideal (see [1]):

Proposition 4.4. *Let $\mathfrak{f} = \mathfrak{f}_1$ be a nonreduced fractional ideal.*

1. *The recursion (4.6) produces a reduced fractional ideal \mathfrak{f}_m equivalent to \mathfrak{f} for some $m \in \mathbb{N}$.*

2. If m in part 1 is minimal, i.e. \mathfrak{f}_m is reduced and \mathfrak{f}_n is nonreduced for $n < m$, then

$$m \leq \max \left\{ 1, \frac{1}{2} \left(5 - \deg(N(\mathfrak{f})) - \frac{1}{4} \deg(\Delta) \right) \right\}.$$

3. If \mathfrak{f} is the product of two reduced ideals and m is as in part 2, then

$$m \leq \frac{3}{8} (\deg(\Delta) + 4).$$

As an aside, we mention the *infrastructure* of the set $\{\mathfrak{f}_1, \mathfrak{f}_2, \dots, \mathfrak{f}_p\}$ of reduced principal fractional ideals. If $\mathfrak{f}_i = (\theta_i)^{-1}$ for $i = 1, 2, \dots, p$, then the *distance* of \mathfrak{f}_i is $\delta_i = \deg(\theta_i)$. From part 3 of Proposition 4.4, a reduced principal ideal \mathfrak{f} can be obtained by applying no more than $3(\deg(\Delta) + 4)/8$ iterations of (4.6) to the initial (generally nonreduced) product ideal $\mathfrak{f}_i \mathfrak{f}_j$. Moreover, $\delta(\mathfrak{f}) = \delta_i + \delta_j + \delta$ with $\delta = O(\deg(\Delta)) = O(\log p)$, so the distance of \mathfrak{f} is within a logarithmically small ‘error’ of where one would expect it to be. As pointed out in section 1, this phenomenon allows for much faster computation of the fundamental unit and other invariants of $K/k(x)$.

The implementation of Algorithm 4.1 raises a number of questions: How large do the degrees of θ , ξ_θ , and η_θ ($\theta \in \{\mu, \nu\}$) and those of their basis coefficients get throughout the algorithm? How often the while loops in steps 3.1 and 4 executed? And how does one determine absolute values of ξ_θ and η_θ , and compute the quantities $\lfloor \xi_\mu / \xi_\nu \rfloor$ in steps 3.2 and 4 as well as $\lfloor \eta_\mu / \eta_\nu \rfloor$ in step 4? These questions will be addressed in the next three sections.

5 Input/Output Sizes in Ideal Basis Reduction

We begin with the following empirical observation; for quadratic integers (as opposed to Puiseux series), this is referred to as the *Gauß-Kuz'min law*. Let $\alpha = \alpha_0 \in k\langle x^{-1} \rangle$ and define $a_i = \lfloor \alpha_i \rfloor \in k[x]$ and $\alpha_{i+1} = (\alpha_i - a_i)^{-1}$ for $i \in \mathbb{N}_0$. Then the a_i ($i \in \mathbb{N}_0$) are the partial quotients in the simple continued fraction expansion of α , and for $i \in \mathbb{N}$, a_i will almost always have very small degree. The quotients $\lfloor \xi_\mu / \xi_\nu \rfloor$ in steps 3.1, 3.2, and the first while loop of step 4 are easily seen to be partial quotients in the simple continued fraction expansion of $\xi_{\mu_0} / \xi_{\nu_0}$ where μ_0 and ν_0 are the inputs of step 3.1 or, if that loop is never entered, of step 3.2; similarly for $\lfloor \eta_\nu / \eta_\mu \rfloor$ in the second while loop of step 4. These quotients will therefore almost always have very small degree, with the possible exception of the very first such partial quotient.

Let $\{1, \mu, \nu\}$ be a reduced basis of some fractional ideal \mathfrak{f} that was computed using Algorithm 4.1. Since $|\eta_\mu| < 1 \leq |\eta_\nu|$, and η_ν and η_μ differ by a factor that is a partial quotient as described above, $|\eta_\nu|$ will usually have quite small degree, and $|\eta_\mu|$ will not be much smaller than 1. By (4.5) and (4.4), $|\xi_\nu| < |\xi_\mu| \leq |\Delta(\mathfrak{f})|^{1/2}$, so usually, $|\xi_\mu|$ will be close to $|\Delta(\mathfrak{f})|^{1/2}$, and since ξ_μ and ξ_ν once again differ by a factor that is a partial quotient in a simple continued fraction expansion, $|\xi_\nu|$ will not be much smaller than $|\xi_\mu|$.

We have the following rigorous bounds on reduced bases:

Proposition 5.1. Let $\{1, \mu, \nu\}$ be a reduced basis of a fractional ideal, where $\mu = (m_0 + m_1\rho + m_2\omega)/d$, $\nu = (n_0 + n_1\rho + n_2\omega)/d$ with $m_0, m_1, m_2, n_0, n_1, n_2 \in k[x]$ and $d = d(\mathfrak{f})$.

1. $\lfloor m_0/d \rfloor = \lfloor m_1\rho/d \rfloor = \lfloor m_2\omega/d \rfloor = 3\lfloor \mu \rfloor$.
2. $|\mu| \leq \max\{q^{-1}, |\Delta(\mathfrak{f})|^{1/2}\}$, $|m_0|, |m_1\rho|, |m_2\omega| \leq \max\{q^{-1}|d|, |\Delta|^{1/2}\}$, $|\nu| \leq \max\{1, q^{-1}|\Delta(\mathfrak{f})|^{1/2}\}$, $|n_1\rho + n_2\omega| < |\Delta|^{1/2}$, $|n_0| \leq \max\{|d|, q^{-1}|\Delta|^{1/2}\}$.
3. If $|\mu| > 1$, then $|\nu| < |\mu| \leq |\Delta(\mathfrak{f})|^{1/2}$, $|m_0| = |m_1\rho| = |m_2\omega| \leq |\Delta|^{1/2}$, $|n_0|, |n_1\rho|, |n_2\omega| < |\Delta|^{1/2}$.

Proof. Part 1 follows immediately from $|\eta_\mu| < 1$ and $|\zeta_\mu| < 1$. For part 2, we note that from (4.2), (4.5), and (4.4) $|\mu| \leq \max\{|\zeta_\mu|, |\xi_\mu|\}$ with $|\zeta_\mu| < 1$ and $|\xi_\mu| = |\Delta(\mathfrak{f})|^{1/2}|\eta_\nu|^{-1} \leq |\Delta(\mathfrak{f})|^{1/2}$. The bounds on $|m_1\rho|$ and $|m_2\omega|$ follow from $|m_1\rho - m_2\omega| = |d\eta_\mu| < |d|$ and $|m_1\rho + m_2\omega| = |d\xi_\mu| \leq |dN(\mathfrak{f})||\Delta|^{1/2} \leq |\Delta|^{1/2}$ by (4.4) and the first two parts of Proposition 3.1. Furthermore, $|m_0| \leq \max\{|d\xi_\nu|, |d\zeta_\nu|\}$. Now by (4.2) $|\nu| \leq \max\{|\zeta_\nu|, |\xi_\nu|\}$ with $|\zeta_\nu| \leq 1$ and $|\xi_\nu| < |\xi_\mu| \leq |\Delta(\mathfrak{f})|^{1/2}$ by (4.4), $|d\xi_\nu| < |d\xi_\mu| \leq |\Delta|^{1/2}$, and $|n_0| \leq \max\{|d\zeta_\nu|, |d\xi_\nu|\} \leq \max\{|d|, q^{-1}|\Delta|^{1/2}\}$. The bounds in part 3 follow from part 1, (4.2), and the fact that $|d| < |\Delta|^{1/2}$ by part 4 of Proposition 3.1.

Note that unfortunately, we have no rigorous upper bound on $|\eta_\nu|$ and hence on $|n_1\rho|$ and $|n_2\omega|$ in the (nonreduced) case where $|\mu| \leq 1$. However, as we saw above, these values will generally not be too large. We proceed to analyze the sizes of the inputs of step 2 of Algorithm 4.1.

Lemma 5.2.

1. Let \mathfrak{f}_1 be a fractional ideal and let $\mathfrak{f}_{n+1} = (\mu_n^{-1})\mathfrak{f}_n$ where $\{1, \mu_n, \nu_n\}$ is a reduced basis of \mathfrak{f}_n ($n \in \mathbb{N}$). Let $\mathfrak{f} = \mathfrak{f}_{n+1} = [1, \mu^{-1}, \nu\mu^{-1}]$ for some $n \in \mathbb{N}$ with $\mu = \mu_n$ and $\nu = \nu_n$. Then

$$\max\left\{|\eta_{\mu^{-1}}|, \frac{|\eta_{\nu\mu^{-1}}|}{|\nu'|}\right\} = \frac{1}{|\mu'|},$$

$$|\xi_{\mu^{-1}}| \leq \frac{1}{\min\{|\mu|, |\mu'|\}}, \quad |\xi_{\nu\mu^{-1}}| \leq \max\left\{\frac{|\nu|}{|\mu|}, \frac{|\nu'|}{|\mu'|}\right\} \leq \frac{|\nu'|}{\min\{|\mu|, |\mu'|\}}.$$

$$|\Delta(\mathfrak{f})|^{1/2} \leq \max\{|\xi_{\mu^{-1}}\eta_{\nu\mu^{-1}}|, |\xi_{\nu\mu^{-1}}\eta_{\mu^{-1}}|\} \leq |\Delta(\mathfrak{f})|^{1/2} \frac{\max\{|\mu|, |\mu'|\}}{|\xi_\mu|}.$$

If \mathfrak{f}_n is reduced, then

$$\max\{|\eta_{\mu^{-1}}|, |\xi_{\mu^{-1}}|\} = \frac{1}{|\mu'|} \leq |\Delta(\mathfrak{f})|^{1/4},$$

$$\max\{|\eta_{\nu\mu^{-1}}|, |\xi_{\nu\mu^{-1}}|\} = \frac{|\nu'|}{|\mu'|} < |\Delta(\mathfrak{f})|^{1/2},$$

$$\max\{|\xi_{\mu^{-1}}\eta_{\nu\mu^{-1}}|, |\xi_{\nu\mu^{-1}}\eta_{\mu^{-1}}|\} = |\Delta(\mathfrak{f})|^{1/2}.$$

If \mathfrak{f}_n is nonreduced and \mathfrak{f}_1 is the product of two reduced fractional ideals, then

$$\max \left\{ |\eta_{\mu^{-1}}|, \frac{|\eta_{\nu\mu^{-1}}|}{|\nu'|} \right\} < |\Delta(\mathfrak{f})\Delta|^{1/4},$$

$$\max \left\{ |\xi_{\mu^{-1}}|, \frac{|\xi_{\nu\mu^{-1}}|}{|\nu'|} \right\} \leq q^{-3}|\Delta(\mathfrak{f})\Delta|^{1/2}.$$

$$\max \left\{ |\xi_{\mu^{-1}}\eta_{\mu^{-1}}|, \frac{|\xi_{\nu\mu^{-1}}\eta_{\nu\mu^{-1}}|}{|\nu'|^2} \right\} \leq q^{-2}|\Delta(\mathfrak{f})\Delta|^{1/2}.$$

2. Let $\{1, \alpha, \beta\}$ be a canonical basis of a fractional ideal \mathfrak{f} . Then

$$\left| \frac{\rho}{d(\mathfrak{f})} \right| \leq |\xi_\alpha|, |\eta_\alpha| \leq |\rho|, \quad \left| \frac{\omega}{d(\mathfrak{f})} \right| \leq \max\{|\xi_\beta|, |\eta_\beta|\} \leq |\omega|.$$

If \mathfrak{f} is reduced, then $|\xi_\alpha|, |\eta_\alpha| > |\omega|^{-1}$, $\max\{|\xi_\beta|, |\eta_\beta|\} > |\rho|^{-1}$.

If \mathfrak{f} is the product of two reduced fractional ideals, then

$$|\xi_\alpha|, |\eta_\alpha| \geq \frac{q^2}{|\Delta|^{1/2}|\omega|}, \quad \max\{|\xi_\beta|, |\eta_\beta|\} \geq \frac{q^2}{|\Delta|^{1/2}|\rho|}.$$

Proof. 1. By (4.3) $|\eta_{\mu^{-1}}| \leq |\mu'|^{-1}$, $|\xi_{\mu^{-1}}| \leq \max\{|\mu|^{-1}, |\mu'|^{-1}\}$, $|\eta_{\nu\mu^{-1}}| \leq |\nu'||\mu'|^{-1}$, and $|\xi_{\nu\mu^{-1}}| \leq \max\{|\nu||\mu|^{-1}, |\nu'||\mu'|^{-1}\}$. Since $|\nu| \leq \max\{1, |\xi_\mu|\} \leq \max\{1, |\mu|\}$, we have $|\nu||\mu|^{-1} \leq \max\{1, |\mu|^{-1}\} \leq |\nu'| \max\{|\mu|^{-1}, |\mu'|^{-1}\}$.

A simple computation reveals that

$$\eta_{\mu^{-1}} = -\frac{\eta_\mu}{\mu'\mu''}, \quad \eta_{\nu\mu^{-1}} = \frac{\eta_\nu\zeta_\mu - \eta_\mu\zeta_\nu}{2\mu'\mu''}.$$

Since $\max\{|\zeta_\mu|, |\eta_\mu|\} = |\mu'|$, one of $\eta_{\mu^{-1}}$ and $\eta_{\nu\mu^{-1}}/\eta_\nu$ has absolute value $|\mu'|^{-1}$.

Now by (4.4) $|\xi_{\mu^{-1}}\eta_{\nu\mu^{-1}} - \xi_{\nu\mu^{-1}}\eta_{\mu^{-1}}| = |\Delta(\mathfrak{f})|^{1/2}$, so $|\Delta(\mathfrak{f})|^{1/2}$ cannot exceed both summands in absolute value. By (4.4), an upper bound on the absolute values of both terms is given by

$$\frac{|\nu'|}{|\mu'|\min\{|\mu|, |\mu'|\}} = \frac{|\Delta(\mathfrak{f}_n)|^{1/2}}{|\mu'\xi_\mu|\min\{|\mu|, |\mu'|\}} = |\Delta(\mathfrak{f})|^{1/2} \frac{\max\{|\mu|, |\mu'|\}}{|\xi_\mu|}$$

since $\Delta(\mathfrak{f}) = N(\mu)^{-2}\Delta(\mathfrak{f}_n)$.

If \mathfrak{f}_n is reduced, then $|\mu| > 1$, so $|\xi_\mu| = \max\{|\mu|, |\mu'|\} = |\mu|$ and $|\nu| \leq \max\{1, |\xi_\nu|\} < |\mu|$. Furthermore,

$$\zeta_{\mu^{-1}} = \frac{\zeta_\mu}{\mu'\mu''}, \quad \zeta_{\nu\mu^{-1}} = \frac{\zeta_\mu\zeta_\nu - 3\eta_\mu\eta_\nu}{2\mu'\mu''}.$$

If $|\zeta_\mu| = |\mu'|$, then $|\eta_{\nu\mu^{-1}}| = |\nu'||\mu'|^{-1}$ and $|\xi_{\mu^{-1}}| = |2\mu^{-1} - \zeta_{\mu^{-1}}| = |\mu'|^{-1}$. If $|\eta_\mu| = |\mu'|$, then $|\eta_{\mu^{-1}}| = |\mu'|^{-1}$ and $|\xi_{\nu\mu^{-1}}| = |2\nu\mu^{-1} - \zeta_{\nu\mu^{-1}}| =$

$|\nu'||\mu'|^{-1}$. Finally, $|\mu'|^2 = |N(\mu)||\mu|^{-1} \geq |N(\mu)||\Delta(\mathfrak{f}_n)|^{-1/2} = |\Delta(\mathfrak{f})|^{-1/2}$ and $|\nu'||\mu'|^{-1} = |\Delta(\mathfrak{f}_n)|^{1/2}|\mu\mu'|^{-1} < |\Delta(\mathfrak{f}_n)|^{1/2}|N(\mu)| = |\Delta(\mathfrak{f})|^{1/2}$.

If \mathfrak{f}_n is nonreduced, then $|\mu| \leq 1$. If \mathfrak{f}_1 is the product of two reduced ideals, then by part 3 of Proposition 3.1, $|\Delta(\mathfrak{f}_n)| \geq |\Delta(\mathfrak{f}_1)| \geq q^4|\Delta|^{-1}$. Then $|\mu'|^2 \geq |N(\mu)| = |\Delta(\mathfrak{f})^{-1}\Delta(\mathfrak{f}_n)|^{1/2} \geq q^2|\Delta(\mathfrak{f})\Delta|^{-1/2}$ and $\min\{|\mu|, |\mu'|\} \geq q^{-1}|N(\mu)| \geq q^3|\Delta(\mathfrak{f})\Delta|^{-1/2}$.

2. Let $\alpha = s^{-1}s'(u + \rho)$, $\beta = s^{-1}s''(v + w\rho + \omega)$. Then $\xi_\alpha = \eta_\alpha = s's^{-1}\rho$, $\xi_\beta = s''s^{-1}(w\rho + \omega)$, $\eta_\beta = s''s^{-1}(w\rho - \omega)$ with $|w| < |s'|$. Since $|s's''| \leq |s| = |d(\mathfrak{f})|$ and $|\rho| \leq |\omega|$, the first set of bounds follows. If \mathfrak{f} is reduced, then $|d(\mathfrak{f})| < |\Delta|^{1/2} = |\rho\omega|$ by part 4 of Proposition 3.1. If \mathfrak{f} is the product of two reduced fractional ideals, then $|d(\mathfrak{f})| \leq |N(\mathfrak{f})|^{-1} \leq q^2|\Delta|$ by parts 2 and 3 of Proposition 3.1.

We point out that in the situation where Algorithm 4.1 is applied to the product \mathfrak{f} of two reduced fractional ideals (as is the case in the infrastructure scenario, for example), the input is a canonical basis and not of the form $\{\mu^{-1}, \nu\mu^{-1}\}$.

We now proceed to investigate the workings of ideal basis reduction in more detail; in particular, we will see how the sizes of the quantities ξ_μ , ξ_ν , η_μ , and η_ν change throughout Algorithm 4.1. We point out that after step 2 of the algorithm, $|\xi_\nu| \leq |\xi_\mu|$ and $|\eta_\nu| \leq |\eta_\mu|$.

Lemma 5.3.

1. In step 3.1 of Algorithm 4.1, ξ_μ and η_μ do not increase in absolute value in the first iteration and decrease in absolute value in each subsequent iteration. ξ_ν and η_ν decrease in absolute value in each iteration. Furthermore, $|\xi_\mu| > |\xi_\nu|$ and $|\eta_\mu| > |\eta_\nu|$ after each iteration.
2. Step 3.2 of Algorithm 4.1 decreases ξ_μ , ξ_ν , and η_μ , but does not decrease η_ν in absolute value. After execution, $|\xi_\mu| > |\xi_\nu|$ and $|\eta_\mu| \leq |\eta_\nu|$.
3. Step 3.3 of Algorithm 4.1 leaves the absolute values of ξ_μ , ξ_ν , and η_ν unchanged, but decreases η_μ in absolute value. After execution, $|\xi_\mu| > |\xi_\nu|$ and $|\eta_\mu| < |\eta_\nu|$.

Proof. Let $\{\alpha, \beta\}$ be the input and $\{\mu, \nu\}$ the output of any iteration of step 3.1, step 3.2, or step 3.3.

Since $|\xi_\nu\eta_\nu| > |\Delta(\mathfrak{f})|^{1/2}$ if and only if $|\xi_\mu/\xi_\nu - \eta_\mu/\eta_\nu| < 1$, or equivalently, if and only if $\lfloor \xi_\mu/\xi_\nu \rfloor = \lfloor \eta_\mu/\eta_\nu \rfloor$, we have in step 3.1

$$\begin{aligned} \xi_\mu &= \xi_\beta, & \xi_\nu &= -\xi_\alpha + \left\lfloor \frac{\xi_\alpha}{\xi_\beta} \right\rfloor \xi_\beta, \\ \eta_\mu &= \eta_\beta, & \eta_\nu &= -\eta_\alpha + \left\lfloor \frac{\eta_\alpha}{\eta_\beta} \right\rfloor \eta_\beta. \end{aligned}$$

Therefore $|\xi_\nu| < |\xi_\beta| = |\xi_\mu|$ and $|\eta_\nu| < |\eta_\beta| = |\eta_\mu|$. From step 2 of the algorithm, in the first iteration $|\xi_\alpha| \geq |\xi_\beta|$ and $|\eta_\alpha| \geq |\eta_\beta|$, so $|\xi_\mu| \leq |\xi_\alpha|$ and $|\eta_\mu| \leq |\eta_\alpha|$. In subsequent iterations, we have $|\xi_\alpha| > |\xi_\beta|$ and $|\eta_\alpha| > |\eta_\beta|$, so $|\xi_\mu| = |\xi_\beta| < |\xi_\alpha|$ and $|\eta_\mu| = |\eta_\beta| < |\eta_\alpha|$.

In step 3.2, the transformations on $\xi_\mu, \xi_\nu, \eta_\mu$ are the same as in step 3.1, so each of these quantities decrease in absolute value, and we still have $|\xi_\mu| \geq |\xi_\nu|$. Furthermore,

$$|\eta_\nu| = \left| \left(-\eta_\alpha + \left\lfloor \frac{\eta_\alpha}{\eta_\beta} \right\rfloor \eta_\beta \right) - \left(\left\lfloor \frac{\eta_\alpha}{\eta_\beta} \right\rfloor - \left\lfloor \frac{\xi_\alpha}{\xi_\beta} \right\rfloor \right) \eta_\beta \right|.$$

The first term in the difference has absolute value less than $|\eta_\beta|$, while the second term is at least $|\eta_\beta|$ in absolute value because $|\lfloor \eta_\alpha/\eta_\beta \rfloor - \lfloor \xi_\alpha/\xi_\beta \rfloor| \geq 1$. So $|\eta_\nu| \geq |\eta_\beta| = |\eta_\mu|$.

In step 3.3, we have $\nu = \beta$, so ξ_ν and η_ν are unchanged. Furthermore, if $a = \text{sgn}(\eta_\alpha \eta_\beta^{-1})$, then $|\xi_\mu| = |\xi_\alpha - a\xi_\beta| = |\xi_\alpha|$ as $|\xi_\alpha| > |\xi_\beta| = |\xi_\nu|$. Finally, since $a = \lfloor \eta_\alpha/\eta_\beta \rfloor$, $|\eta_\mu| < |\eta_\beta| = |\eta_\nu|$.

Analogous results hold for step 4 of Algorithm 4.1:

Lemma 5.4.

1. In the first loop of step 4 of Algorithm 4.1, ξ_μ and ξ_ν decrease in absolute value in each iteration, while η_μ and η_ν increase in absolute value in each iteration.
2. In the seconds loop of step 4 of Algorithm 4.1, ξ_μ and ξ_ν increase in absolute value in each iteration, while η_μ and η_ν decrease in absolute value in each iteration.
3. Throughout step 4, $|\xi_\mu| > |\xi_\nu|$ and $|\eta_\mu| < |\eta_\nu|$. At most one of the while loops in step 4 is entered, and after the last iteration of either of the loops, $|\xi_\mu| > |\xi_\nu|$ and $|\eta_\mu| < 1 \leq |\eta_\nu|$.

The previous two lemmata show that $|\eta_\nu|$ takes on its largest value throughout the algorithm either after step 3.2 or after the first loop of step 4 if that value is less than 1 after step 3.2. Since in both cases $|\eta_\nu \xi_\mu| = |\Delta(f)|^{1/2}$, and we generally at least expect $|\xi_\mu| \geq |d(f)|^{-1}$, we usually have by parts 1 and 2 of Proposition 3.1 $|\eta_\nu| \leq |d||\Delta(f)|^{1/2} \leq |\Delta|^{1/2}$ for this maximal value.

6 Complexity of Ideal Basis Reduction

We now investigate how often each of the while loops in the basis reduction algorithm.

Proposition 6.1. *Let $f = [1, \mu, \nu]$ where μ, ν are the inputs of Algorithm 4.1. Assume that $|\xi_\mu| \geq |\xi_\nu|$ and $|\eta_\mu| \geq |\eta_\nu|$, so step 2 has been executed. Denote by r , s , and t the number of iterations of step 3.1, the first loop in step 4, and the second loop in step 4, respectively. Then*

$$r \leq \max \left\{ 0, \frac{1}{2} \left(\deg(\xi_\nu \eta_\nu) - \frac{1}{2} \deg(\Delta(f)) + 1 \right) \right\},$$

$$r + s \leq \max \{ 0, \deg(\xi_\nu) - \frac{1}{2} \deg(\Delta(f)) \}, \quad r + t \leq \max \{ 0, \deg(\eta_\nu) + 1 \}.$$

Proof. Let $\{\mu_0, \nu_0\}$ be the first input and $\{\mu_i, \nu_i\}$ the output after iteration i ($1 \leq i \leq r$) of step 3.1. From part 1 of Lemma 5.3 $|\xi_{\nu_i}| < |\xi_{\nu_{i-1}}|$ and $|\eta_{\nu_i}| < |\eta_{\nu_{i-1}}|$, so inductively $|\xi_{\nu_i}| \leq q^{-i}|\xi_\nu|$ and $|\eta_{\nu_i}| \leq q^{-i}|\eta_\nu|$ for $1 \leq i \leq r$. Then $|\Delta(\mathfrak{f})|^{1/2} \leq q^{-1}|\xi_{\nu_{r-1}}\eta_{\nu_{r-1}}| \leq q^{1-2r}|\xi_\nu\eta_\nu|$, so $q^r \leq (|\xi_\nu\eta_\nu||\Delta(\mathfrak{f})|^{-1/2}q)^{1/2}$.

Again, let $\{\mu_0, \nu_0\}$ be the first input and $\{\mu_i, \nu_i\}$ the output after iteration i ($1 \leq i \leq s$) of the first loop of step 4. Then $|\eta_{\nu_0}| < 1$. Analogous to the previous part, we infer from Lemma 5.4 that $|\eta_{\nu_i}| \geq q^i|\eta_{\nu_0}|$ for $1 \leq i \leq s$, and $|\eta_{\nu_{s-1}}| < 1 \leq |\eta_{\nu_s}|$. Then $1 \geq q|\eta_{\nu_{s-1}}| \geq q^s|\eta_{\nu_0}|$. Here, ν_0 is the ν value output by step 3.3 and hence by 3.2 (since 3.3 leaves it unchanged). Thus, the corresponding η_ν is the quantity $\eta_{\nu_{r+1}}$, where we interpret step 3.2 as the $(r+1)$ -st iteration of the loop in step 3.1. Now $|\eta_{\nu_{r+1}}| = |\Delta(\mathfrak{f})|^{1/2}|\xi_{\nu_r}|^{-1} \geq q^r|\Delta(\mathfrak{f})|^{1/2}|\xi_\nu|^{-1}$. Thus, $q^{r+s} \leq |\xi_\nu||\Delta(\mathfrak{f})|^{-1/2}$.

In the second loop of step 4 of Algorithm 4.1, we have $|\eta_{\mu_i}| \leq q^{-i}|\eta_{\mu_0}|$ for $1 \leq i \leq t$, and $|\eta_{\mu_t}| < 1 \leq |\eta_{\mu_{t-1}}|$. Then $1 \leq |\eta_{\nu_{t-1}}| \leq q^{-(t-1)}|\eta_{\mu_0}|$, where $|\mu_0|$ is μ value output by step 3.3. The corresponding $|\eta_{\mu_0}|$ is at most equal to $|\eta_{\mu_{r+1}}|$. Then $|\eta_{\mu_{r+1}}| = |\eta_{\nu_r}| \leq q^{-r}|\eta_\nu|$, so $q^{r+t} \leq q|\eta_\nu|$.

Corollary 6.2. *Let r , s , and t be as in Lemma 5.2. Let \mathfrak{f} be the input ideal and $\{1, \mu, \nu\}$ the input basis of Algorithm 4.1. Assume that $|\xi_\mu| \geq |\xi_\nu|$ and $|\eta_\mu| \geq |\eta_\nu|$, so step 2 has been executed.*

1. Suppose $\mathfrak{f} = \mathfrak{f}_{n+1}$ for some $n \in \mathbb{N}$, where \mathfrak{f}_1 is a fractional ideal, $\mathfrak{f}_{n+1} = (\mu_n^{-1})\mathfrak{f}_n$ with $\{1, \mu_n, \nu_n\}$ a reduced basis of \mathfrak{f}_n ($n \in \mathbb{N}$).

If \mathfrak{f}_n is reduced, then $r = s = 0$, $t \leq \frac{1}{4}\deg(\Delta(\mathfrak{f})) + 1$.

If \mathfrak{f}_n is nonreduced and \mathfrak{f}_1 is the product of two reduced fractional ideals, then

$$r \leq \frac{1}{4}\deg(\Delta) - \frac{1}{2}, \quad r + s \leq \frac{1}{2}\deg(\Delta) - 3, \quad r + t \leq \frac{1}{4}\deg(\Delta(\mathfrak{f})\Delta) + 1.$$

2. Suppose $\{1, \mu, \nu\}$ is a canonical basis of \mathfrak{f} . Then

$$r \leq \frac{1}{2}(\deg(d(\mathfrak{f}))+1), \quad r + s \leq \max\{0, \deg(d(\mathfrak{f})) - \deg(\omega)\}, \quad r + t \leq \deg(\rho) + 1.$$

If $\mathfrak{f} = \mathcal{O}$, i.e. $\nu = \rho$ and $\mu = \omega$, then $r = s = 0$, $t \leq \deg(\rho) + 1$.

If \mathfrak{f} is reduced, then $r \leq \frac{1}{4}\deg(\Delta)$, $r + s < \deg(\rho)$, $r + t \leq \deg(\rho) + 1$.

If \mathfrak{f} is the product of two reduced fractional ideals, then

$$r < \frac{1}{2}\deg(\Delta), \quad r + s \leq \frac{1}{2}\deg(\Delta) + \deg(\rho) - 2, \quad r + t \leq \deg(\rho) + 1.$$

Proof. Part 1 follows directly from the bounds in Lemma 5.2. For part 2, let $\{1, \alpha, \beta\}$ be a canonical basis of \mathfrak{f} with $\alpha = s's^{-1}\rho$ and $\beta = s''s^{-1}(w\rho + \omega)$. Since $|\rho| \leq |\omega|$, $|\xi_\alpha\eta_\alpha||\Delta(\mathfrak{f})|^{-1/2} \leq |s'\rho||s''\omega|^{-1} \leq |s| = |d(\mathfrak{f})|$, $|\xi_\alpha||\Delta(\mathfrak{f})|^{-1/2} = |s||s''\omega|^{-1} \leq |d(\mathfrak{f})||\omega|^{-1}$, and $|\eta_\alpha| \leq |\rho|$. Once again by Proposition 3.1, $|d(\mathfrak{f})| < |\Delta|^{1/2}$ if \mathfrak{f} is reduced and $|d(\mathfrak{f})| \leq q^{-2}|\Delta|$ if \mathfrak{f} is the product of two reduced ideals. If $\nu = \rho$ and $\mu = \omega$, then $|\xi_\nu\eta_\nu| = |\rho|^2 \leq |\Delta|^{1/2} = |\Delta(\mathcal{O})|^{1/2}$ and $|\xi_\nu| < |\Delta|^{1/2}$.

Corollary 6.2 reveals that if the input ideal \mathfrak{f} of Algorithm 4.1 is either equal to \mathcal{O} (with basis $\{1, \rho, \omega\}$), or is of the form $\mathfrak{f} = \mathfrak{f}_{n+1} = (\mu_n^{-1})\mathfrak{f}_n$ where \mathfrak{f}_n is reduced and $\{1, \mu_n, \nu_n\}$ is a reduced basis of \mathfrak{f}_n , then step 3.1, the first while loop in step 4, and step 6 can be omitted. This is the case, in particular, if the regulator or the fundamental unit of $K/k(x)$ are computed by generating the recursion $\mathfrak{f}_{n+1} = (\mu_n^{-1})\mathfrak{f}_n$ with $\mathfrak{f}_1 = \mathfrak{f}_{p+1} = \mathcal{O}$.

Algorithm 6.3. (*Basis Reduction, Input Ideal of Special Form*)

Input: $\tilde{\mu}, \tilde{\nu}$ where $\{1, \tilde{\mu}, \tilde{\nu}\}$ is a basis of some fractional ideal \mathfrak{f} . Here, $\{\tilde{\mu}, \tilde{\nu}\} = \{\rho, \omega\}$ or $\{\tilde{\mu}, \tilde{\nu}\} = \{\phi^{-1}, \theta\phi^{-1}\}$ where $\{1, \phi, \theta\}$ is a reduced basis of a reduced fractional ideal.

Output: μ, ν where $\{1, \mu, \nu\}$ is a reduced basis of \mathfrak{f} .

Algorithm:

1. Set $\mu = \tilde{\mu}, \nu = \tilde{\nu}$.

2. If $|\xi_\mu| < |\xi_\nu|$ or if $|\xi_\mu| = |\xi_\nu|$ and $|\eta_\mu| < |\eta_\nu|$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

3. If $|\eta_\mu| \geq |\eta_\nu|$

3.1. Replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 0 & 1 \\ -1 & \lfloor \xi_\mu / \xi_\nu \rfloor \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

3.2. If $|\eta_\mu| = |\eta_\nu|$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 1 & -\text{sgn}(\eta_\mu \eta_\nu^{-1}) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

4. While $|\eta_\mu| \geq 1$, replace

$$\begin{pmatrix} \mu \\ \nu \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} \lfloor \eta_\nu / \eta_\mu \rfloor & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \nu \end{pmatrix}.$$

5. Replace μ by $\mu - \lfloor \zeta_\mu \rfloor / 2$ and ν by $\nu - \lfloor \zeta_\nu \rfloor / 2$.

7 Precision Required for Ideal Basis Reduction

When computing absolute values as well as integer parts of quotients as required in basis reduction algorithm, the relevant quantities of the form $b\rho \pm c\omega$ need to be approximated to sufficient “precision” with a Puiseux series in $k\langle x^{-1} \rangle$ that is truncated at some suitable negative power of x . Our numerical experiments in [2] show that increasing the precision or even using variable precision does not have a significant impact on the running time of the algorithm; for example, a reduction in precision from $\deg(D)$ to $\deg(D)/2$ made a difference of only 5-10 percent in computation time. Nevertheless, it is desirable to have a lower bound

on the minimal precision required; in [2], where we implemented Algorithm 4.1 for reduced ideals only, we relied exclusively on heuristics and numerical evidence in determining our precision.

We define a *relative approximation of precision* $n \in \mathbb{N}_0$ to an element $\alpha = \sum_{i=-m}^{\infty} a_i x^{-i} \in k\langle x^{-1} \rangle$ to be $\hat{\alpha}_n = \sum_{i=-m}^{n-\deg(\alpha)} a_i x^{-i}$. Then $|1 - \hat{\alpha}/\alpha| < q^{-n}$, or equivalently, $|\alpha - \hat{\alpha}| < q^{\deg(\alpha)-n}$. To approximate a quantity of the form $\theta = b\rho + c\omega$ with $b, c \in k(x)$, such as $\xi_\mu, \xi_\nu, \eta_\mu$, and η_ν , we generate relative approximations $\hat{\rho}_n$ and $\hat{\omega}_n$ of sufficient precision n to ρ and ω , respectively, and approximate θ by $\hat{\theta} = b\hat{\rho}_n + c\hat{\omega}_n$. $\hat{\rho}_n$ is precomputed by explicitly extracting a cube root of $D \in k[x]$ so that the coefficients of $x^{\deg(D)/3}, \dots, x, 1, x^{-1}, \dots, x^{n-\deg(D)/3}$ are correct, and $\hat{\omega}_n$ is given by the following lemma.

Lemma 7.1. *Let $\hat{\rho}_n$ be a relative approximation of precision n to ρ . Then $\hat{\omega}_n = \lfloor x^{n-\deg(\omega)} \hat{\rho}_n^2 / H \rfloor x^{\deg(\omega)-n}$ is a relative approximation of precision n to ω .*

Here, it is a simple matter to verify that $|1 - \omega_n/\hat{\omega}_n| < q^{-n}$. Henceforth, we denote by $\hat{\rho}_n$ and $\hat{\omega}_n$ relative approximations of some precision $n \in \mathbb{N}$ to ρ and ω , respectively. For $\theta = a + b\rho + c\omega$ with $a, b, c \in k(x)$, we set

$$\hat{\theta} = a + b\hat{\rho}_n + c\hat{\omega}_n, \quad \hat{\xi}_\theta = b\hat{\rho}_n + c\hat{\omega}_n, \quad \hat{\eta}_\theta = b\hat{\rho}_n - c\hat{\omega}_n, \quad \hat{\zeta}_\theta = 2a - b\hat{\rho}_n - c\hat{\omega}_n.$$

The following lemma gives lower bounds on the precision required to compute absolute values and integer parts of certain Puiseux series correctly.

Lemma 7.2. *Let $\theta, \phi \in k(x)$.*

1. *If $m \in \mathbb{Z}$ and $q^{n+m} \geq \max\{|\xi_\theta|, |\eta_\theta|\}$, then $|\xi_\theta| = q^m$ if and only if $|\hat{\xi}_\theta| = q^m$, $|\xi_\theta| \leq q^m$ if and only if $|\hat{\xi}_\theta| \leq q^m$, and $|\xi_\theta| < q^m$ if and only if $|\hat{\xi}_\theta| < q^m$.*
2. *If $q^n \geq \max\left\{1, \left|\frac{\eta_\theta}{\xi_\theta}\right|\right\}$, then $|\xi_\theta| = |\hat{\xi}_\theta|$.*
3. *If $q^n \geq \max\{|\xi_\theta|, |\eta_\theta|\}$, then $\lfloor \theta \rfloor = \lfloor \hat{\theta} \rfloor$ and $\lfloor \zeta_\theta \rfloor = \lfloor \hat{\zeta}_\theta \rfloor$.*
4. *If $|\xi_\theta| \geq |\xi_\phi|$ and $q^n \geq \max\left\{1, \left|\frac{\xi_\theta}{\xi_\phi}\right|, \left|\frac{\eta_\theta}{\xi_\phi}\right|, \left|\frac{\xi_\theta \eta_\phi}{\xi_\phi^2}\right|\right\}$, then $\left\lfloor \frac{\xi_\theta}{\xi_\phi} \right\rfloor = \left\lfloor \frac{\hat{\xi}_\theta}{\hat{\xi}_\phi} \right\rfloor$.*

Proof. If $\theta = a + b\rho + c\omega$ with $a, b, c \in k[x]$, then

$$\begin{aligned} |\theta - \hat{\theta}| &= |\xi_\theta - \hat{\xi}_\theta| = |\zeta_\theta - \hat{\zeta}_\theta| \\ &= |b(\rho - \hat{\rho}_n) + c(\omega - \hat{\omega}_n)| < \max\{|b\rho|, |c\omega|\} q^{-n} = \max\{|\xi_\theta|, |\eta_\theta|\} q^{-n}. \end{aligned}$$

This immediately yields parts 1–3. For part 4, we have

$$\frac{\hat{\xi}_\theta}{\hat{\xi}_\phi} = \frac{\xi_\theta}{\xi_\phi} + \frac{\xi_\theta(\xi_\phi - \hat{\xi}_\phi)}{\xi_\phi \hat{\xi}_\phi} + \frac{\hat{\xi}_\theta - \xi_\theta}{\hat{\xi}_\phi}.$$

Suppose that $|\xi_\theta| \geq |\xi_\phi|$ and $q^n \geq \max\{1, |\xi_\theta/\xi_\phi|, |\eta_\theta/\xi_\phi|, |\xi_\theta \eta_\phi/\xi_\phi^2|\}$. Then

$$\left| \frac{\xi_\theta(\xi_\phi - \hat{\xi}_\phi)}{\xi_\phi \hat{\xi}_\phi} \right| < \frac{|\xi_\theta|}{|\xi_\phi|^2} \max\{|\xi_\phi|, |\eta_\phi|\} q^{-n} \leq 1;$$

similarly, $|(\hat{\xi}_\theta - \xi_\theta)/\hat{\xi}_\phi| < 1$. So $\lfloor \xi_\theta/\xi_\phi \rfloor = \lfloor \hat{\xi}_\theta/\hat{\xi}_\phi \rfloor$.

We are now able to give lower bounds on n for the different steps of Algorithm 4.1. We consider a precision of n to be sufficient if in any identity or condition on a quantity θ , θ can be replaced by a relative approximation $\hat{\theta}$ of precision n to θ . For example, n is sufficient for step 3.1 of Algorithm 4.1 if $|\xi_\nu \eta_\nu| > |\Delta(f)|^{1/2}$ exactly if $|\hat{\xi}_\nu \hat{\eta}_\nu| > |\Delta(f)|^{1/2}$ and if $[\xi_\mu / \xi_\nu] = [\hat{\xi}_\mu / \hat{\xi}_\nu]$ in every iteration of the loop.

Lemma 7.3. *Let f be the input ideal and $\{1, \mu, \nu\}$ the input basis of Algorithm 4.1. Define $\{\alpha, \beta\} = \{\gamma, \delta\} = \{\mu, \nu\}$ such that $|\xi_\alpha| \geq |\xi_\beta|$ and $|\eta_\gamma| \geq |\eta_\delta|$. Let r , s , and t be as in Proposition 6.1. Then a precision of n is sufficient for*

1. step 2 and the if condition at the start of step 3 if $q^n \geq \max \left\{ \left| \frac{\eta_\gamma}{\xi_\alpha} \right|, \left| \frac{\xi_\alpha}{\eta_\gamma} \right| \right\}$;
2. step 3.1 if $q^n \geq \max \left\{ \left| \frac{\xi_\alpha}{\xi_\beta} \right|, \left| \frac{\xi_\alpha}{\eta_\gamma} \right|, \left| \frac{\eta_\gamma}{\xi_\beta} \right|, \frac{|\xi_\beta|^2}{|\Delta(f)|^{1/2}}, \frac{|\xi_\beta \eta_\delta|}{|\Delta(f)|^{1/2}}, q^{l_r-2} \frac{|\eta_\delta|^2}{\Delta(f)^{1/2}} \right\}$
where $q^{l_r} = \left| \frac{\xi_{\mu_r}}{\xi_{\nu_r}} \right|$ for $r \geq 1$;
3. step 3.2 if $q^n \geq \max \left\{ \left| \frac{\xi_\alpha}{\xi_\beta} \right|, \left| \frac{\eta_\gamma}{\xi_\beta} \right|, \left| \frac{\xi_\alpha \eta_\delta}{\xi_\beta^2} \right|, q^{l_r}, q^{2l_r-2} \frac{|\eta_\delta|^2}{\Delta(f)^{1/2}} \right\}$;
4. step 3.3 if $q^n \geq \max \left\{ 1, \frac{|\xi_\beta|^2}{|\Delta(f)|^{1/2}} \right\}$;
5. the first while loop of step 4 if $q^n \geq \max \left\{ |\Delta(f)|^{1/2}, |\xi_\beta|, q^l, \frac{q^{2l_{s-1}-2}}{|\Delta(f)|^{1/2}} \right\}$ where $q^l = \max_{0 \leq i \leq s-1} \left\{ \left| \frac{\xi_{\mu_i}}{\xi_{\nu_i}} \right| \right\}$ and $q^{l_{s-1}} = \left| \frac{\xi_{\mu_{s-1}}}{\xi_{\nu_{s-1}}} \right|$;
6. the second while loop of step 4 if $q^n \geq \max \left\{ q^m, q^{m_t} |\Delta(f)|^{1/2} \right\}$ where $q^m = \max_{0 \leq j \leq t-1} \left\{ \left| \frac{\eta_{\nu_j}}{\eta_{\mu_j}} \right| \right\}$ and $q^{m_t} = \left| \frac{\eta_{\nu_t}}{\eta_{\mu_t}} \right|$;
7. steps 5 and 6 if $q^n \geq \max \{q^{m_t}, |\Delta(f)|^{1/2}\}$.

Proof. We use the results of Lemma 7.2 and the same notation as in the proof of Proposition 6.1. We only prove parts 1–3 and part 5; the other parts follow analogously.

1. Since $q^n \geq \max \{1, |\eta_\alpha / \xi_\alpha|\}$, $|\xi_\alpha| = |\hat{\xi}_\alpha|$, and since $q^n \geq \max \{|\xi_\beta / \xi_\alpha|, |\eta_\beta / \xi_\alpha|\}$, $|\xi_\nu| < |\xi_\mu|$ if and only if $|\hat{\xi}_\nu| < |\hat{\xi}_\mu|$. Finally, $q^n \geq \max \{|\eta_\delta / \eta_\gamma|, |\xi_\delta / \xi_\gamma|\}$ implies $|\eta_\mu| < |\eta_\nu|$ if and only if $|\hat{\eta}_\mu| < |\hat{\eta}_\nu|$ and $|\eta_\mu| \geq |\eta_\nu|$ if and only if $|\hat{\eta}_\mu| \geq |\hat{\eta}_\nu|$.
2. We have $\alpha = \gamma = \mu_0$ and $\beta = \delta = \nu_0$, $|\xi_{\nu_i}| \leq |\xi_\beta|$, $|\eta_{\nu_i}| \leq |\eta_\delta|$, and $|\xi_{\nu_i} \eta_{\nu_i}| > |\Delta(f)|^{1/2} \geq |\xi_{\nu_r} \eta_{\nu_r}|$ for $0 \leq i \leq r-1$. Furthermore, $|\xi_{\mu_i} / \xi_{\nu_i}| = |\eta_{\mu_i} / \eta_{\nu_i}|$, so $|\eta_{\nu_i} / \xi_{\nu_i}| = |\eta_\delta / \xi_\beta| = |\eta_\gamma / \xi_\alpha|$ for $0 \leq i \leq r-1$.

Hence, since $q^n \geq \max\{1, |\eta_\gamma/\xi_\alpha|, |\xi_\alpha/\eta_\gamma|\}$, $|\xi_{\nu_i}| = |\hat{\xi}_{\nu_i}|$, and $|\eta_{\nu_i}| = |\hat{\eta}_{\nu_i}|$ for $0 \leq i \leq r-1$. Also, if $r \geq 1$, then

$$\left| \frac{\eta_{\nu_r}}{\xi_{\nu_r}} \right| = q^{l_r} \left| \frac{\eta_{\nu_r} \eta_{\nu_{r-1}}}{\xi_{\nu_{r-1}} \eta_{\nu_{r-1}}} \right| \leq q^{l_r-2} \frac{|\eta_\delta|^2}{|\Delta(\mathfrak{f})|^{1/2}},$$

so $|\xi_{\nu_r}| = |\hat{\xi}_{\nu_r}|$. Furthermore, since

$$q^n \geq \max \left\{ \frac{|\xi_\beta \eta_\delta|}{|\Delta(\mathfrak{f})|^{1/2}}, \frac{|\xi_\beta|^2}{|\Delta(\mathfrak{f})|^{1/2}} \right\} \geq \frac{|\xi_{\nu_i}|}{|\Delta(\mathfrak{f})|^{1/2}} \max\{|\eta_{\nu_i}|, |\xi_{\nu_i}|\},$$

then $|\eta_{\nu_i}| \leq |\Delta(\mathfrak{f})|^{1/2}/|\xi_{\nu_i}|$ if and only if $|\hat{\eta}_{\nu_i}| \leq |\Delta(\mathfrak{f})|^{1/2}/|\xi_{\nu_i}|$ for $0 \leq i \leq r$. Finally, if $r \geq i \geq 1$, then $|\xi_{\mu_i}/\xi_{\nu_i}| < |\xi_{\nu_{i-1}} \eta_{\nu_i}|/|\Delta(\mathfrak{f})|^{1/2} < |\xi_\beta \eta_\delta|/|\Delta(\mathfrak{f})|^{1/2}$ and $|\eta_{\mu_i}/\xi_{\nu_i}| < |\eta_\delta|^2/|\Delta(\mathfrak{f})|^{1/2}$. Also, $|\xi_{\mu_i} \eta_{\nu_i}/\xi_{\nu_i}|^2 = |\eta_{\nu_i}/\xi_{\nu_i}|$ for $0 \leq i \leq r-1$. Hence, $q^n \geq \max\{|\xi_\alpha/\xi_\beta|, |\eta_\gamma/\xi_\beta|, |\xi_\beta \eta_\delta|/|\Delta(\mathfrak{f})|^{1/2}, |\eta_\delta|^2/|\Delta(\mathfrak{f})|^{1/2}\}$ implies $\lfloor \xi_{\mu_i}/\xi_{\nu_i} \rfloor = \lfloor \hat{\xi}_{\mu_i}/\hat{\xi}_{\nu_i} \rfloor$ for $0 \leq i \leq r-1$.

3. If $r \geq 1$, then $|\xi_{\mu_r}/\xi_{\nu_r}| = q^{l_r}$,

$$\left| \frac{\eta_{\mu_r}}{\xi_{\nu_r}} \right| = q^{l_r} \left| \frac{\eta_{\nu_{r-1}}^2}{\xi_{\nu_{r-1}} \eta_{\nu_{r-1}}} \right| < q^{l_r} \frac{|\eta_\delta|^2}{|\Delta(\mathfrak{f})|^{1/2}},$$

$$\left| \frac{\xi_{\mu_r} \eta_{\nu_r}}{\xi_{\nu_r}^2} \right| = q^{2l_r} \left| \frac{\eta_{\nu_r} \eta_{\nu_{r-1}}}{\xi_{\nu_{r-1}} \eta_{\nu_{r-1}}} \right| \leq q^{2l_r-2} \frac{|\eta_\delta|^2}{|\Delta(\mathfrak{f})|^{1/2}}.$$

5. We have $|\eta_{\nu_i}| < 1 \leq |\eta_{\nu_s}|$, so $|\xi_{\mu_i}| > |\Delta(\mathfrak{f})|^{1/2} \geq |\xi_{\mu_s}|$ for $0 \leq i \leq s-1$. Since $q^n \geq \max\{1, |\xi_\beta|\} \geq \max\{|\eta_{\nu_i}|, |\xi_{\nu_i}| : 0 \leq i \leq s-1\}$, $|\eta_{\nu_i}| < 1$ if and only if $|\hat{\eta}_{\nu_i}| < 1$. Also $|\xi_{\nu_s}/\eta_{\nu_s}| \leq |\Delta(\mathfrak{f})|^{1/2}$, so $q^n \geq \max\{1, |\Delta(\mathfrak{f})|^{1/2}\}$ yields $|\eta_{\nu_s}| = |\hat{\eta}_{\nu_s}|$.

Now $|\xi_{\mu_i}/\xi_{\nu_i}| \leq q^l$ for $0 \leq i \leq s-1$, and for $0 \leq i \leq s-2$:

$$\left| \frac{\eta_{\mu_i}}{\xi_{\nu_i}} \right| < \left| \frac{\eta_{\nu_i}}{\xi_{\mu_{i+1}}} \right| < \frac{1}{|\Delta(\mathfrak{f})|^{1/2}}, \quad \left| \frac{\xi_{\mu_i} \eta_{\nu_i}}{\xi_{\nu_i}^2} \right| = \frac{|\Delta(\mathfrak{f})|^{1/2}}{\xi_{\mu_{i+1}}^2} < \frac{1}{|\Delta(\mathfrak{f})|^{1/2}}$$

and

$$\left| \frac{\eta_{\mu_{s-1}}}{\xi_{\nu_{s-1}}} \right| \leq \frac{q^{l_{s-1}-2}}{|\xi_{\mu_{s-1}}|} \leq \frac{q^{l_{s-1}-3}}{|\Delta(\mathfrak{f})|^{1/2}},$$

$$\left| \frac{\eta_{\nu_{s-1}} \xi_{\mu_{s-1}}}{\xi_{\nu_{s-1}}^2} \right| = \frac{|\Delta(\mathfrak{f})|^{1/2}}{|\xi_{\nu_{s-1}}|^2} = q^{2l_{s-1}} \frac{|\Delta(\mathfrak{f})|^{1/2}}{|\xi_{\mu_{s-1}}|^2} \leq \frac{q^{2l_{s-1}-2}}{|\Delta(\mathfrak{f})|^{1/2}}.$$

It follows that $\lfloor \xi_{\mu_i}/\xi_{\nu_i} \rfloor = \lfloor \hat{\xi}_{\mu_i}/\hat{\xi}_{\nu_i} \rfloor$ for $0 \leq i \leq s-1$.

Corollary 7.4. *Let \mathfrak{f} be the input ideal and $\{1, \mu, \nu\}$ the input basis of Algorithm 4.1 or Algorithm 6.3. Define $\{\alpha, \beta\} = \{\gamma, \delta\} = \{\mu, \nu\}$ such that $|\xi_\alpha| \geq |\xi_\beta|$ and $|\eta_\gamma| \geq |\eta_\delta|$. Let l , l_r , l_{s-1} , m , and m_t be as in Lemma 7.3.*

1. If $q^n \geq \max \left\{ \left| \xi_\beta \right|, \left| \frac{\xi_\alpha}{\xi_\beta} \right|, \left| \frac{\eta_\gamma}{\xi_\beta} \right|, \left| \frac{\xi_\alpha}{\eta_\gamma} \right|, \left| \frac{\xi_\alpha \eta_\delta}{\xi_\beta^2} \right|, \frac{|\xi_\beta|^2}{|\Delta(\mathfrak{f})|^{1/2}}, \frac{|\xi_\beta \eta_\delta|}{|\Delta(\mathfrak{f})|^{1/2}}, \frac{q^{l_r-2} |\eta_\delta|^2}{|\Delta(\mathfrak{f})|^{1/2}}, \frac{q^{2l_{s-1}-2}}{|\Delta(\mathfrak{f})|^{1/2}}, q^l, q^m, q^{m_t}, q^{m_t} |\Delta(\mathfrak{f})|^{1/2}, \right\}$, then a precision of n is sufficient for Algorithm 4.1.
2. If $q^n \geq \max \left\{ \left| \frac{\xi_\alpha}{\xi_\beta} \right|, \left| \frac{\eta_\gamma}{\xi_\beta} \right|, \left| \frac{\xi_\alpha}{\eta_\gamma} \right|, \left| \frac{\xi_\alpha \eta_\delta}{\xi_\beta^2} \right|, \frac{|\xi_\beta|^2}{|\Delta(\mathfrak{f})|^{1/2}}, q^m, q^{m_t} |\Delta(\mathfrak{f})|^{1/2} \right\}$, then a precision of n is sufficient for Algorithm 6.3.

We point out that the values q^l , q^{l_r} , $q^{l_{s-1}}$, q^m , and q^{m_t} are almost always very small. In general, we expect the case where \mathfrak{f} is the product of two reduced ideals to require the highest precision, since in this case, $|N(\mathfrak{f})|^{-1}$ (and hence the upper bound on $|d(\mathfrak{f})|$ by part 2 of Proposition 3.1) is largest. Even in this situation, it is very likely that the required precision is not too large, say no more than $\deg(\Delta)$; however, only numerical experiments will tell. The scenario of Algorithm 6.3 requires significantly less precision: here, we expect $\deg(\Delta)/2$ to be sufficient, and this bound is supported by numerical evidence (see [2]).

8 Conclusion and Outlook

We have provided a complete analysis of the algorithm for computing a reduced basis of a fractional ideal in a purely cubic function field of unit rank 1. The number of iterations of each while loop of the algorithm is bounded by a fraction of $\deg(\Delta)$. The quantities $|\xi_\mu|$, $|\xi_\nu|$, $|\eta_\mu|$, and $|\eta_\nu|$ appear not to grow too large throughout our computations; in fact, we expect the bounds of Lemma 5.2 to significantly exceed the actual sizes of these quantities. Finally, the precision required to compute absolute values and quotients appears to be a fraction of $\deg(\Delta)$ as well.

As mentioned in section 1, our two algorithms serve two purposes. If Algorithm 6.3 is repeatedly applied, starting and terminating with $\mathfrak{f} = \mathcal{O}$, it generates all the reduced principal fractional ideals in \mathcal{O} and thus produces the fundamental unit and/or the regulator of $K/k(x)$ as illustrated in [2]. Algorithm 4.1 can be used to determine from a given nonreduced fractional ideal an equivalent reduced one. In particular, if the input ideal is the product of two reduced principal ideals, then the infrastructure of the set of reduced fractional principal ideals guarantees that the method finds a reduced principal fractional ideal “close” to the product ideal very quickly, namely after at most $3(\deg(\Delta) + 4)/8$ applications of Algorithm 4.1. This phenomenon allows for a rapid movement through this set, thereby speeding up regulator and fundamental unit computation significantly. The technique can be extended to yield the ideal class number of $K/k(x)$ and hence the order of the group of k -rational points on the Jacobian of K . Work on this problem is currently in progress.

If $q \equiv -1 \pmod{3}$, then a representation of unit rank 1 can always be achieved for any purely cubic extension $K/k(x)$ by applying a simple change

of variable; in particular, any purely cubic extension of unit rank 0 (i.e. when $\deg(D)$ is not a multiple of 3) can always be converted to one of unit rank 1 over the same field of rational functions $k(x)$. The methods outlined above can also undoubtedly be generalized to arbitrary cubic function fields of unit rank 1; once again, this is currently being explored. In addition, we are in the process of investigating the case of even characteristic. It remains to be seen which elements of Algorithms 4.1 and 6.3 (if any) are of use in cubic extensions of unit rank 2, and to what extent our techniques can be extended to unit rank 1 extensions of degree higher than 3. Much of the reduction theory remains valid here, but Algorithm 4.1 needs to be replaced by an entirely different reduction procedure.

References

1. R. Scheidler, *Ideal Arithmetic and Infrastructure in Purely Cubic Function Fields*. To appear in *J. Th. Nombr. Bordeaux*.
2. R. Scheidler and A. Stein, Voronoi's Algorithm in Purely Cubic Congruence Function Fields of Unit Rank 1. To appear in *Math. Comp.* **69** (2000), 1245–1266.
3. A. Stein and H. C. Williams, Some methods for evaluating the regulator of a real quadratic function field. *Exp. Math.* **8** (1999), 119–133.
4. G. F. Voronoi, *On a Generalization of the Algorithm of Continued Fractions* (in Russian). Doctoral Dissertation, University of Warsaw (Poland) 1896.

Factorization in the Composition Algebras

Derek A. Smith*

Lafayette College, Easton, PA 18042
`smithder@lafayette.edu`

Abstract. Let \mathcal{O} be a maximal arithmetic in one of the four (non-split) composition algebras over \mathbb{R} , and let $[\rho] = mn$ be the norm of an element ρ in \mathcal{O} . Rehm [14] describes an algorithm for finding all factorizations of ρ as $\rho = \alpha\beta$, where $[\alpha] = m$, $[\beta] = n$ and $(m, n) = 1$. Here, we extend the algorithm to general ρ , m , and n , providing precise geometrical configurations for the sets of left- and right-hand divisors.

1 Introduction

A composition algebra over \mathbb{R} for the bilinear form

$$[x, y] = x_1y_1 + \cdots + x_ny_n,$$

where we define the (squared) norm of an element x as $[x] = [x, x]$, is a not-necessarily-associative division algebra satisfying the composition law

$$[xy] = [x][y]$$

for all $x, y \in \mathbb{R}^n$. Hurwitz [8] proved that composition algebras occur only in dimensions $n = 1, 2, 4$, and 8 ; the algebras, unique up to isomorphism, consist of \mathbb{R} , \mathbb{C} , the quaternions \mathbb{H} , and the octonions \mathbb{O} . Frobenius [7] had earlier shown that \mathbb{R} , \mathbb{C} , and \mathbb{H} are the only associative finite-dimensional division algebras over \mathbb{R} . The octonions are not associative, but they are an alternative algebra, meaning that the left and right alternative laws

$$x^2y = x(xy) \text{ and } yx^2 = (yx)x$$

hold for all $x, y \in \mathbb{O}$. By a theorem of Artin (see [11]), any subalgebra of an alternative algebra generated by two elements is associative.

Multiplication in the octonions can be described quite easily using the following sets of coordinates. Let any element $x \in \mathbb{O}$ be an \mathbb{R} -linear combination of eight orthogonal unit vectors $1 = i_\infty, i_0, i_1, \dots, i_6$. For $t = 0, \dots, 6$, define multiplication among $1, i_t, i_{t+1}, i_{t+3}$ (subscripts taken mod 7) to coincide with the multiplication of the basis elements $1, i, j, k$ of the quaternions, \mathbb{H} :

$$\begin{aligned} i^2 &= j^2 = k^2 = -1 \\ ij &= k \quad ji = -k \end{aligned}$$

* I wish to thank J. H. Conway and Princeton University for their support during my graduate work. The results of this paper are taken from my doctoral dissertation.

Since \mathbb{C} can be defined as having basis elements 1 and i , we see the containments $\mathbb{R} \subset \mathbb{C} \subset \mathbb{H} \subset \mathbb{O}$.

Each composition algebra \mathcal{C} contains sets of elements that are *arithmetics* in the sense of Dickson [4] and Lamont [9]. An arithmetic A is a subset of \mathcal{C} which contains 1; is closed under addition, subtraction, and multiplication; and is such that each element $\alpha \in A$ satisfies $[\alpha] \in \mathbb{Z}$ and $2[\alpha, 1] \in \mathbb{Z}$. A is said to be a *maximal* arithmetic if it is not contained in any other arithmetic of \mathcal{C} .

Define the *naive* arithmetic $N_{\mathcal{C}}$ as the one whose elements are simply \mathbb{Z} -linear combinations of 1 and the imaginary units given above for $\mathcal{C} = \mathbb{C}, \mathbb{H}$ and \mathbb{O} ; and let \mathcal{O} represent any arithmetic of \mathcal{C} containing $N_{\mathcal{C}}$. In the remainder of this paper, the term arithmetic will refer only to such \mathcal{O} . For $\mathcal{C} = \mathbb{R}$ and \mathbb{C} , \mathcal{O} is the set of rational integers \mathbb{Z} and Gaussian integers $\{a + bi \mid a, b \in \mathbb{Z}\}$, respectively. For $\mathcal{C} = \mathbb{H}$, up to isomorphism there is one arithmetic properly containing $N_{\mathbb{H}}$, namely Hurwitz' integral quaternions

$$\{a + bi + cj + dk \mid \text{either } a, b, c, d \in \mathbb{Z}, \text{ or } a, b, c, d \in \mathbb{Z} + \frac{1}{2}\}.$$

For $\mathcal{C} = \mathbb{O}$, there are four non-isomorphic \mathcal{O} : $N_{\mathbb{O}} \subset \mathcal{O}^1 \subset \mathcal{O}^2 \subset \mathcal{O}^3$ [9], [15]. The maximal arithmetic \mathcal{O}^3 can be described in terms of the coordinates given above for \mathbb{O} :

$$\mathcal{O}^3 = \{a_{\infty}i_{\infty} + a_0i_0 + \cdots + a_6i_6 \mid \text{each } a_t \in \mathbb{Z}/2, \text{ and } \{a_t\} \cap \mathbb{Z} \in S\},$$

where S consists of the subsets of $\{a_{\infty}, a_0, \dots, a_6\}$ whose indices are taken from $\{\emptyset, 0124, 0235, 0346, \infty 045, 0156, \infty 026, \infty 013\}$ and the complements of these indices in $\infty 0123456$. Geometrically, \mathcal{O}^3 is similar to the E_8 lattice [2].

The problem of finding the factorizations of a given $\rho \in \mathcal{O}$ as $\rho = \alpha\beta$ for $\alpha, \beta \in \mathcal{O}$ and fixed $m = [\alpha]$ and $n = [\beta]$ has a long history. Factorization results for \mathcal{O} in \mathbb{R} , \mathbb{C} , and \mathbb{H} are classical (see [4]). However, the methods of associative number theory are not well-suited to $\mathcal{O} \subset \mathbb{O}$ since, for instance, every one-sided ideal in \mathcal{O}^3 is in fact two-sided and generated by a rational integer [1], [10].

We now summarize what is known for the four non-isomorphic $\mathcal{O} \subset \mathbb{O}$. Rankin [13], in a study of multiplicative functions, gives the number of factorizations in $N_{\mathbb{O}}$ in the two cases $(m, n) = 1$ and $m = p$, $n = p^k$, where p is a prime. Pall and Taussky [12], using results of Estes and Pall [5] on the genera of certain octonary quadratic forms, determine the factorizations in $N_{\mathbb{O}}$ for general m and n . Feaux and Hardy [6] then extend their work to \mathcal{O}^1 , \mathcal{O}^2 , and \mathcal{O}^3 . Unfortunately, these results, although “constructive,” do not lead readily to a geometric understanding of the sets of divisors.

Recently Rehm [14] produced an algorithm that finds all factorizations of ρ in the maximal octonion arithmetic \mathcal{O}^3 when $(m, n) = 1$. In this paper, we show that methods can be extended to general ρ , m , and n in the maximal arithmetic of any composition algebra, providing precise geometrical configurations for the sets of left- and right-hand divisors.

2 The Algorithm

Let \mathcal{O} be a maximal arithmetic of a composition algebra \mathcal{C} containing $N_{\mathcal{C}}$, and let $\mathcal{O}_m \subset \mathcal{O}$ consist of the elements of norm m . We start with $\rho_1 \in \mathcal{O}_{m_0 m_1}$, where $m_0 \geq m_1 > 0$. We wish to find the set $L_{m_0}(\rho_1) \subset \mathcal{O}_{m_0}$ of left-hand divisors of ρ_1 of norm m_0 .

Define the *conjugate* $\bar{\alpha}$ of an element α to be $\bar{\alpha} = 2[\alpha, 1] - \alpha$. From the shape of Voronoi cell of \mathcal{O} (see Coxeter [3] for $\mathcal{O} = \mathcal{O}^3 \subset \mathbb{O}$), there exists a pair $\{\gamma_1, \rho_2\} \subset \mathcal{O}$ such that

$$\rho_1 = \gamma_1 m_1 + \overline{\rho_2}, \text{ where } 0 \leq [\overline{\rho_2}] \leq \frac{[m_1]}{2} = \frac{m_1^2}{2}.$$

m_1 divides $[\rho_2]$, since m_1 divides every term on the right-hand side of

$$[\rho_2] = [\overline{\rho_2}] = [\rho_1 - \gamma_1 m_1] = [\rho_1] + m_1^2 [\gamma_1] - m_1 (2[\rho_1, \gamma_1]).$$

Let m_2 be such that $[\rho_2] = m_1 m_2$. Then $m_1 > \frac{m_1}{2} \geq m_2 \geq 0$. If $m_2 \neq 0$, we can repeat the arguments above with ρ_2 and m_2 , leading to ρ_3 and m_3 ; and so on. At some point we must reach an $m_{N+1} = 0$, and thus $\rho_{N+1} = 0$, since $m_i > m_{i+1} \geq 0$ for any $m_i > 0$ and $i \geq 1$. Thus, we obtain a finite collection of elements in \mathcal{O} whose relationships are summarized in Figure 1.

$$\begin{array}{lll} \rho_1 = \gamma_1 m_1 + \overline{\rho_2} & [\rho_1] = m_0 m_1 & m_0 \geq m_1 \\ \rho_2 = \gamma_2 m_2 + \overline{\rho_3} & [\rho_2] = m_1 m_2 & m_1 > m_2 \\ \vdots & \vdots & \vdots \\ \rho_{N-1} = \gamma_{N-1} m_{N-1} + \overline{\rho_N} & [\rho_{N-1}] = m_{N-2} m_{N-1} & m_{N-2} > m_{N-1} \\ \rho_N = \gamma_N m_N & [\rho_N] = m_{N-1} m_N & m_{N-1} > m_N > 0 \end{array}$$

Fig. 1. A “Euclidean algorithm” in \mathcal{O} .

Now, let μ_N be any element of \mathcal{O}_{m_N} . Since any subalgebra of \mathcal{O} generated by two elements is associative, we may write

$$\rho_N = \gamma_N m_N = \gamma_N (\mu_N \overline{\mu_N}) = (\gamma_N \mu_N) \overline{\mu_N}.$$

Set $\mu_{N-1} = \gamma_N \mu_N$, so that μ_{N-1} is a left-hand divisor of ρ_N of norm m_{N-1} . Then $\overline{\mu_{N-1}}$ is a right-hand divisor of both $\overline{\rho_N}$ and m_{N-1} , and thus also of ρ_{N-1} , since

$$\begin{aligned} \rho_{N-1} &= \gamma_{N-1} m_{N-1} + \overline{\rho_N} = (\gamma_{N-1} \mu_{N-1}) \overline{\mu_{N-1}} + \mu_N \overline{\mu_{N-1}} \\ &= (\gamma_{N-1} \mu_{N-1} + \mu_N) \overline{\mu_{N-1}}. \end{aligned}$$

Setting $\mu_{N-2} = \gamma_{N-1} \mu_{N-1} + \mu_N$, we obtain a left-hand divisor of ρ_{N-1} of norm m_{N-2} . We can continue this procedure until we arrive at a left-hand divisor μ_0 of ρ_1 of norm m_0 . Figure 2 summarizes this process, which can be thought of as

$$\begin{array}{lll}
\rho_N = \mu_{N-1} \overline{\mu_N} & \mu_{N-1} = \gamma_N \mu_N & [\mu_N] = m_N \\
\rho_{N-1} = \mu_{N-2} \overline{\mu_{N-1}} & \mu_{N-2} = \gamma_{N-1} \mu_{N-1} + \mu_N & [\mu_{N-1}] = m_{N-1} \\
\rho_{N-2} = \mu_{N-3} \overline{\mu_{N-2}} & \mu_{N-3} = \gamma_{N-2} \mu_{N-2} + \mu_{N-1} & [\mu_{N-2}] = m_{N-2} \\
& \vdots & \vdots \\
\rho_2 = \mu_1 \overline{\mu_2} & \mu_1 = \gamma_2 \mu_2 + \mu_3 & [\mu_1] = m_1 \\
\rho_1 = \mu_0 \overline{\mu_1} & \mu_0 = \gamma_1 \mu_1 + \mu_2 & [\mu_0] = m_0
\end{array}$$

Fig. 2. Factoring the ρ_j .

tracing the left-hand side of Figure 1 from bottom to top, factoring along the way.

We remark that the multiplication of the algebra is not used in an essential way in Figure 1. Moreover, products involving triples of elements in Figure 2 occur only within associative subalgebras.

3 The Configurations of the Divisor Sets

We now determine $L_{m_0}(\rho_1)$.

Lemma 1. *Let $\gamma = \alpha\beta = \alpha'\beta'$, where $[\alpha] = [\alpha'] \neq 0$ and $[\beta] = [\beta'] \neq 0$. Then the angle θ_a between α and α' is equal to the angle θ_b between β and β' .*

Proof: Taking the inner product of γ with $\alpha\beta'$, we obtain

$$[\alpha][\beta, \beta'] = [\alpha\beta, \alpha\beta'] = [\gamma, \alpha\beta'] = [\alpha'\beta', \alpha\beta'] = [\alpha', \alpha][\beta'],$$

which yields

$$\cos \theta_a = \frac{[\alpha, \alpha']}{[\alpha]} = \frac{[\beta, \beta']}{[\beta]} = \cos \theta_b. \square$$

To initiate the procedure presented in Figure 2, take μ_N to be any member of \mathcal{O}_{m_N} . Denoting geometrical similarity by \sim , we obtain the following sequence of similarities from Lemma 1 by alternatively setting γ equal to ρ_i and m_i for appropriate i :

$$\mathcal{O}_{m_N} = \{\overline{\mu_N}\} \sim \{\mu_{N-1}\} \sim \{\overline{\mu_{N-1}}\} \sim \cdots \sim \{\overline{\mu_1}\} \sim \{\mu_0\} = L_{m_0}(\rho_1).$$

The final equality follows from Lemma 1 and the relationships among the ρ_i , since distinct $\mu \in L_{m_0}(\rho_1)$ correspond to distinct $\mu' \in L_{m_N}(\rho_N) = \mathcal{O}_{m_N}$. Note that we also now know the set of right-hand divisors of ρ_1 of norm m_1 : $R_{m_1}(\rho_1) = \{\overline{\mu_1}\}$.

We still have to determine m_N . Let $\gcd(\eta_1, \dots, \eta_k)$ denote the greatest common rational integer divisor of η_1, \dots, η_k in \mathcal{O} .

Lemma 2. Let $d_i = \gcd(\rho_i, m_{i-1}, m_i)$ for $1 \leq i \leq N$. Then $d_i = d_{i+1}$ for $1 \leq i < N$.

Proof: First, see that $d_i \mid \overline{\rho_{i+1}} = \rho_i - \gamma_i m_i$ since d_i divides ρ_i and m_i , so $d_i \mid \rho_{i+1}$. It divides m_i by definition. Finally, $d_i \mid m_{i+1}$ since d_i divides each term in the last line of

$$\begin{aligned} m_{i+1} &= \frac{[\rho_{i+1}]}{m_i} = \left(\frac{1}{m_i}\right)([\rho_i] + m_i^2[\gamma_i] - m_i(2[\gamma_i, \rho_i])) \\ &= m_{i-1} + m_i[\gamma_i] - 2[\gamma_i, \rho_i]. \end{aligned}$$

Thus, $d_i \mid d_{i+1}$. By a similar argument, $d_{i+1} \mid d_i$ as well, so $d_i = d_{i+1}$. \square Note that $m_N = \gcd(\rho_N, m_{N-1}, m_N)$ since m_N divides both ρ_N and m_{N-1} . Thus, Lemma 2 implies that $m_N = \gcd(\rho_1, m_0, m_1)$.

In all of these discussions, we could just as well have computed the set of right-hand divisors of ρ_1 of norm m_0 . Thus, we conclude with

Main Theorem. Let $\rho \in \mathcal{O}$ have norm mn , and let d be the greatest common rational integer divisor of ρ , m , and n . Then the sets of right- and left-hand divisors of ρ of norm m are geometrically similar to \mathcal{O}_d .

References

- [1] D. Allcock. Ideals in the integral octaves. *Journal of Algebra*, to appear.
- [2] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups*. Springer-Verlag, second edition, 1993.
- [3] H. S. M. Coxeter. Integral Cayley numbers. *Duke Math Journal*, 13:561–578, 1946.
- [4] L. E. Dickson. *Algebras and Their Arithmetics*. Univ. Chicago Press, 1923.
- [5] D. Estes and G. Pall. Modules and rings in the Cayley algebra. *Journal of Number Theory*, 1:163–178, 1969.
- [6] C. J. Feaux and J. T. Hardy. Factorization in certain Cayley rings. *Journal of Number Theory*, 7:208–220, 1975.
- [7] G. F. Frobenius. Ueber lineare Substitution und bilineare Formen. *J. Reine Angewandte Mathematik*, 84(1), 1878.
- [8] A. Hurwitz. Über die Komposition der quadratischen Formen von beliebig vielen Variablen. *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, pages 309–316, 1898.
- [9] P. J. C. Lamont. Arithmetics in Cayley's algebra. *Proceedings of the Glasgow Mathematical Association*, 6(1):99–106, 1963.
- [10] K. Mahler. On ideals in the Cayley-Dickson algebra. *Proceedings of the Royal Irish Academy*, 48(5):123–133, 1942.
- [11] R. Moufang. Zur Struktur von Alternativkörpern. *Mathematische Annalen*, 110:416–430, 1934.
- [12] G. Pall and O. Taussky. Factorization of Cayley numbers. *Journal of Number Theory*, 2:74–90, 1970.
- [13] R. A. Rankin. A certain class of multiplicative functions. *Duke Mathematical Journal*, 13(1):281–306, 1946.
- [14] H. P. Rehm. Prime factorization of integral Cayley octaves. *Annales de la Faculté des Sciences de Toulouse*, 2(2):271–289, 1993.
- [15] F. van der Blij and T. A. Springer. The arithmetics of octaves and of the group G_2 . *Nederl. Akad. Wetensch. Indag. Math.*, 21:406–418, 1959.

A Fast Algorithm for Approximately Counting Smooth Numbers

Jonathan P. Sorenson

Computer Science Department
Butler University
4600 Sunset Avenue
Indianapolis, Indiana 46208, USA
sorenson@butler.edu
<http://euclid.butler.edu/~sorenson>

Abstract. Let $\Psi(x, y)$ denote the number of integers $\leq x$ that are composed entirely of primes bounded by y . We present an algorithm for estimating the value of $\Psi(x, y)$ with a running time roughly proportional to \sqrt{y} . Our algorithm is a modification of an algorithm by Hunter and Sorenson that is based on a theorem of Hildebrand and Tenenbaum. This previous algorithm ran in time roughly proportional to y .

1 Introduction

Let $\Psi(x, y)$ denote the number of integers $\leq x$ that have prime divisors $\leq y$. The running times of many integer factoring and discrete logarithm algorithms make use of this function. As a number of important cryptography protocols rely on the difficulty of either integer factoring or the discrete logarithm problem, it is important for the security of such schemes to have good estimates for $\Psi(x, y)$ (see for example [10,13]). Mathematicians have studied the behavior of this function and obtained a number of estimates for it [5,6,7,8,11,12].

Until recently, the standard way to estimate $\Psi(x, y)$ computationally was to use the estimate $\Psi(x, y) \approx x\rho(\log x / \log y)$ [6], as Dickman's function $\rho(u)$ is relatively easy to compute. However, Hunter and Sorenson [9] showed that a theorem of Hildebrand and Tenenbaum [7] gives a much better approximation to $\Psi(x, y)$ in practice, and can be computed using a number of floating point operations that is roughly proportional to y . We refer to this method as *Algorithm HT*.

In this paper, we show how to modify Algorithm HT to improve its running time to roughly \sqrt{y} floating point operations. We pay for this drastic improvement in a slightly larger error, and we need to assume the Riemann Hypothesis to show the error is not excessive. We also present the results of some experiments comparing our improved *Algorithm HT-fast* to the original.

Several other algorithms for estimating $\Psi(x, y)$ deserve mention. Bernstein has presented an algorithm for computing $\Psi(x, y)$ exactly [3], and another algorithm that gives rigorous upper and lower bounds on $\Psi(x, y)$ [4]. Also, several other algorithms are mentioned in [9], although they are not original to that paper.

2 Background

Before we discuss our improvements to Algorithm HT, we need to review some background material. We begin by reviewing Algorithm HT.

2.1 Algorithm HT

First, we introduce some notation and state the theorem upon which the algorithm is based. Define $u = \log x / \log y$.

Let $\bar{u} := \bar{u}(x, y) = \min\{\log x, y\} / \log y = \min\{u, y / \log y\}$.

Define

$$\begin{aligned}\zeta(s, y) &:= \prod_{p \leq y} (1 - p^{-s})^{-1}; \\ \phi(s, y) &:= \log \zeta(s, y); \\ \phi_k(s, y) &:= \frac{d^k}{ds^k} \phi(s, y) \quad (k \geq 1); \\ HT(x, y, s) &:= \frac{x^s \zeta(s, y)}{s \sqrt{2\pi \phi_2(s, y)}}.\end{aligned}$$

Let $\alpha = \alpha(x, y)$ be the unique solution to the equation

$$\phi_1(\alpha, y) + \log x = 0. \tag{1}$$

Theorem 1. *We have*

$$\Psi(x, y) = HT(x, y, \alpha(x, y))(1 + O(1/\bar{u})) \tag{2}$$

uniformly for $2 \leq y \leq x$.

For the proof of this theorem, see Hildebrand and Tenenbaum [7].

Algorithm HT then proceeds as follows:

1. Compute a list of primes up to y using a sieve (see, for example, [14]).
2. Set $\alpha_0 := \log(1 + y/(5 \log x)) / \log y$.
3. Using α_0 as a starting point, find a solution α' to (1) via Newton's method.
Stop when $|\alpha - \alpha'| \leq \min\{0.0001, 1/(\bar{u} \log x)\}$.
4. Output $HT(x, y, \alpha')$.

In theory, in Step 3 above a preliminary search by bisection is required to guarantee a running time of $O(y \{ \frac{\log \log x}{\log y} + \frac{1}{\log \log y} \})$ floating point operations. In practice, Newton's method converges quite nicely after only a few iterations. In [9], it is proved that $HT(x, y, \alpha') = HT(x, y, \alpha)(1 + O(1/\bar{u}))$.

In Steps 2 and 3, the following formulas are used:

$$\begin{aligned}\zeta(s, y) &:= \prod_{p \leq y} (1 - p^{-s})^{-1}; \\ \phi_1(s, y) &:= \sum_{p \leq y} -\frac{\log p}{p^s - 1}; \\ \phi_2(s, y) &:= \sum_{p \leq y} \frac{p^s (\log p)^2}{(p^s - 1)^2}.\end{aligned}$$

We can now briefly explain what our improvements are to this algorithm. The idea is, instead of finding the primes up to y and using them to evaluate ζ , ϕ_1 , and ϕ_2 , we only use the primes up to roughly \sqrt{y} , and then approximate these functions using the prime number theorem. We will, however, need the Riemann Hypothesis in order to bound our error.

2.2 Computation Model

We measure the complexity of our algorithm by counting the number of *floating point* operations. Such operations include addition, subtraction, multiplication, division, comparisons, exponentiation, and taking logarithms of real numbers. We also include array indexing and branching as basic operations. In practice, we used 80-bit floating point numbers.

2.3 Notation

p always denotes a prime number, and sums over p are always sums over primes. $\pi(x)$ denotes the number of primes up to x . For positive functions f and g , we write $f(n) = O(g(n))$ if there exists an absolute constant $c > 0$ such that $f(n) < c \cdot g(n)$ for all n sufficiently large. $f(n) \ll g(n)$ means $f(n) = O(g(n))$.

2.4 Approximating Sums of Primes

We define

$$\text{li}(x) := \int_0^x \frac{1}{\log t} dt$$

where the point at $t = 1$ is omitted. Let $\pi(x) = \text{li}(x) + \epsilon(x)$ (the prime number theorem). By assuming the validity of the Riemann Hypothesis, we can take $\epsilon(t) = O(\sqrt{t} \log t)$. We make frequent use of the following lemma.

Lemma 2. *Let $f(p)$ be a continuously differentiable function on an open interval containing $[2, \infty)$, and let $2 \leq z \leq y$. Then we have*

$$\sum_{z < p \leq y} f(p) = \int_z^y \frac{f(t)}{\log t} dt + f(y)\epsilon(y) - f(z)\epsilon(z) - \int_z^y \epsilon(t)f'(t)dt.$$

For details, see Bach and Shallit [2], Section 2.7 and Theorem 8.3.3.

3 Algorithm HT-Fast

We begin by defining the following three functions:

$$\begin{aligned}
 A(s, y, z) &:= \left(\prod_{p \leq z} (1 - p^{-s})^{-1} \right) \exp \left[\sum_{k=1}^{\lfloor \log y/k \rfloor} \frac{1}{k} (\text{li}(y^{1-ks}) - \text{li}(z^{1-ks})) \right]; \\
 B(s, y, z) &:= \sum_{p \leq z} \frac{\log p}{p^s - 1} + \sum_{k=1}^{\lfloor \log y/k \rfloor} \frac{1}{1-ks} (y^{1-ks} - z^{1-ks}); \\
 C(s, y, z) &:= \sum_{p \leq z} \frac{p^s (\log p)^2}{(p^s - 1)^2} + \frac{z \log z}{s(z^s - 1)} - \frac{y \log y}{s(y^s - 1)} \\
 &\quad + \frac{1}{s(1-ks)} \sum_{k=1}^{\lfloor \log y/k \rfloor} \left[y^{1-ks} \left(1 + \log y - \frac{1}{1-ks} \right) \right. \\
 &\quad \left. - z^{1-ks} \left(1 + \log z - \frac{1}{1-ks} \right) \right]. \\
 HT_f(x, y, z, s) &:= \frac{x^s A(s, y, z)}{s \sqrt{2\pi C(s, y, z)}}.
 \end{aligned}$$

To evaluate $\text{li}(x)$, we use standard techniques for the exponential integral: either 5.1.55 or 5.1.11 (truncated) as appropriate from Abramowitz and Stegun [1]. The time to compute this is only $O(1)$ operations. Thus, given a list of primes up to z , the three functions A , B , and C can be computed in $O(\pi(z) + \log y/s)$ operations. This is significantly smaller than the $O(\pi(y))$ operations to compute ζ , ϕ_1 , or ϕ_2 .

Let $\delta > 0$, and assume that $5 \log x < y^{1/2+\delta}$. Our new algorithm is as follows:

1. Set $z := \min\{y, \max\{1000, 5\sqrt{y}\}\}$.
2. Compute a list of primes up to z .
3. Set $\alpha_0 := \log(1 + y/(5 \log x))/\log y$.
4. Using α_0 as a starting point, find a solution α_f to (1) via Newton's method, substituting $B(s, y, z)$ for $-\phi_1(s, y)$. Stop when:
 $|\alpha - \alpha_f| \leq \min\{0.000001, 0.1/(\bar{u} \log x)\}$ and $|\log x - B(\alpha_f, y, z)| \leq 1$.
5. Output $HT_f(x, y, z, \alpha_f)$.

In the three theorems that follow, we show that our three new functions reasonably approximate $\zeta(s, y)$, $\phi_1(s, y)$, and $\phi_2(s, y)$. After that, we show that the root α_f found in Step 4 will in fact be a good approximation to α .

Theorem 3. *Let $\delta > 0$ such that $1 > s \geq 1/2 + \delta$, and let $z \rightarrow \infty$ such that $2 \leq z \leq y$. Assuming the validity of the Riemann Hypothesis, we have*

$$\zeta(s, y) = A(s, y, z) \left(1 + O\left(\frac{\log z}{z^\delta}\right) \right).$$

Proof. We have

$$\begin{aligned}\zeta(s, y) &= \prod_{p \leq z} (1 - p^{-s})^{-1} \prod_{z < p \leq y} (1 - p^{-s})^{-1} \\ &= \prod_{p \leq z} (1 - p^{-s})^{-1} \exp \left(\sum_{z < p \leq y} -\log(1 - p^{-s}) \right).\end{aligned}$$

Focusing on the sum inside the exponential, we have

$$\begin{aligned}\sum_{z < p \leq y} -\log(1 - p^{-s}) &= - \sum_{z < p \leq y} \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} (-p^{-s})^k \\ &= \sum_{k=1}^{\infty} \sum_{z < p \leq y} \frac{p^{-ks}}{k}\end{aligned}$$

where we have used the expansion $\log(1 + x) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} x^k$ for $|x| < 1$. Next, we approximate using Lemma 2:

$$\sum_{z < p \leq y} \frac{p^{-ks}}{k} = \int_z^y \frac{t^{-ks}}{k \log t} dt + \frac{y^{-ks}}{k} \epsilon(y) - \frac{z^{-ks}}{k} \epsilon(z) + \int_z^y \epsilon(t) st^{-ks-1} dt.$$

With our assumption that $s \geq 1/2 + \delta$ and that $\epsilon(t) = O(\sqrt{t} \log t)$, we can bound our three error terms by $O(\log z / (kz^{k\delta}))$. Substituting $v = t^{1-ks}$, the integral simplifies to $(\text{li}(y^{1-ks}) - \text{li}(z^{1-ks})) / k$.

Finally, observe that we can truncate the infinite sum at $k = \lfloor (\log y) / s \rfloor$ without any significant effect on the error, as the sum converges geometrically. \square

Note that our requirement in the algorithm that $5 \log x < y^{1/2+\delta}$ forces $1/2 + \delta \leq \alpha_0$, which guarantees the condition on s during the algorithm, as we will always have $\alpha_0 \leq s$. The requirement that $s < 1$ is not very restrictive, as $\alpha \leq 1 + O(1/\log y)$ was proven in Lemma 4.1 of [8].

Next, we address approximating $\phi_1(s, y)$.

Theorem 4. *Under the same hypotheses as the previous theorem, we have*

$$-\phi_1(s, y) = B(s, y, z) + O\left(\frac{(\log z)^2}{z^\delta}\right).$$

Proof. We have

$$-\phi_1(s, y) = \sum_{p \leq z} \frac{\log p}{p^s - 1} + \sum_{z < p \leq z} \frac{\log p}{p^s - 1}.$$

Approximating the second term using Lemma 2, we have

$$\begin{aligned} \sum_{z < p \leq z} \frac{\log p}{p^s - 1} &= \int_z^y \frac{1}{t^s - 1} dt + \frac{\log y}{y^s - 1} \epsilon(y) - \frac{\log z}{z^s - 1} \epsilon(z) \\ &\quad - \int_z^y \epsilon(t) \left(\frac{1}{t(t^s - 1)} - \frac{st^{s-1} \log t}{(t^s - 1)^2} \right) dt. \end{aligned}$$

Making use of the fact that $s \geq 1/2 + \delta$ and that by the Riemann Hypothesis we have $\epsilon(t) = O(\sqrt{t} \log t)$, we see that the three error terms are bounded by $O((\log z)^2/z^\delta)$.

To evaluate the integral, we observe that

$$\begin{aligned} \int_z^y \frac{1}{t^s - 1} dt &= \int_z^y \left[\sum_{k=1}^{\infty} t^{-ks} \right] dt \\ &= \sum_{k=1}^{\infty} \int_z^y t^{-ks} dt \\ &= \sum_{k=1}^{\infty} \frac{1}{1 - ks} (y^{1-ks} - z^{1-ks}). \end{aligned}$$

As in the previous theorem, we truncate the infinite sum at $k = \lfloor (\log y)/s \rfloor$. \square

And now the theorem for $\phi_2(s, y)$:

Theorem 5. *Under the same hypotheses as the previous two theorems, we have*

$$\phi_2(s, y) = C(s, y, z) + O\left(\frac{(\log z)^3}{z^\delta}\right).$$

Proof. We have

$$\phi_2(s, y) = \sum_{p \leq z} \frac{p^s (\log p)^2}{(p^s - 1)^2} + \sum_{z < p \leq z} \frac{p^s (\log p)^2}{(p^s - 1)^2}.$$

Approximating the second term using Lemma 2, we have

$$\begin{aligned} \sum_{z < p \leq z} \frac{p^s (\log p)^2}{(p^s - 1)^2} &= \int_z^y \frac{t^s \log t}{(t^s - 1)^2} dt + \frac{y^s (\log y)^2}{(y^s - 1)^2} \epsilon(y) - \frac{z^s (\log z)^2}{(z^s - 1)^2} \epsilon(z) \\ &\quad - \int_z^y \epsilon(t) \left(\frac{d}{dt} \frac{t^s (\log t)^2}{(t^s - 1)^2} \right) dt. \end{aligned}$$

Making use of the fact that $s \geq 1/2 + \delta$ and that by the Riemann Hypothesis we have $\epsilon(t) = O(\sqrt{t} \log t)$, we see that the three error terms are bounded by $O((\log z)^3/z^\delta)$.

To evaluate the integral, we integrate by parts to obtain

$$\int_z^y \frac{t^s \log t}{(t^s - 1)^2} dt = \frac{z \log z}{s(z^s - 1)} - \frac{y \log y}{s(y^s - 1)} + \int_z^y \frac{1 + \log t}{s(t^s - 1)} dt.$$

Using the expansion

$$\frac{1}{t^s - 1} = \sum_{k=1}^{\infty} \frac{1}{t^{ks}}$$

and interchanging summations we then obtain the following for the third term above:

$$\frac{1}{s} \sum_{k=1}^{\infty} \left[\int_z^y t^{-ks} dt + \int_z^y t^{-ks} \log t dt \right].$$

Integrating, we obtain

$$\frac{1}{s} \sum_{k=1}^{\infty} \left[\frac{y^{1-ks}}{1-ks} - \frac{z^{1-ks}}{1-ks} + \frac{y^{1-ks}(\log y - 1/(1-ks))}{1-ks} - \frac{z^{1-ks}(\log z - 1/(1-ks))}{1-ks} \right].$$

We truncate the sum as before. \square

Finally, our theorem that shows that α_f will approximate α .

Theorem 6. *Under the same hypotheses as the previous three theorems, if $|\log x - B(s, y, z)| \ll 1$, then $|s - \alpha| \ll 1/(\log x \log y)$.*

Proof. Our proof has two phases: first we show that $|\alpha - s| \ll 1/(\log y)^2$, and then we use this to deduce the theorem.

We begin by observing that $\phi_2(s, y), \phi_2(\alpha, y) > \phi_2(1, y) \gg (\log y)^2$. This uses the fact that $s, \alpha < 1$ and that ϕ_2 is decreasing, and then we estimate the sum using the prime number theorem.

Next, by the mean value theorem there exists a real number t between s and α such that

$$\phi_1(s, y) = \phi_1(\alpha, y) + (s - \alpha)\phi_2(t, y).$$

By definition, $-\phi_1(\alpha, y) = \log x$; by Theorem 4 we have $-\phi_1(s, y) = B(s, y, z) + o(1)$. Combining this information we have

$$\begin{aligned} |s - \alpha| &= \frac{|\phi_1(s, y) - \phi_1(\alpha, y)|}{\phi_2(t, y)} \\ &\ll \frac{|B(s, y, z) - \log x + O(1)|}{(\log y)^2} \ll \frac{1}{(\log y)^2}. \end{aligned}$$

That completes Step 1.

For Step 2, we can now assume $|\alpha - s| \ll 1/(\log y)^2$, and note that we have $\bar{u} = u = \log x / \log y$ as a consequence of our condition that $s > 1/2$. We take a

Taylor series expansion of ϕ_1 about α to obtain

$$\begin{aligned}\phi_1(s, y) &= \phi_1(\alpha, y) + \sum_{k=0}^{\infty} \frac{(s - \alpha)^{k+1}}{(k+1)!} \phi_{k+2}(\alpha, y) \\ &= \phi_1(\alpha, y) + (s - \alpha) \left(\phi_2(\alpha, y) + \sum_{k=1}^{\infty} \frac{(s - \alpha)^k}{(k+1)!} \phi_{k+2}(\alpha, y) \right).\end{aligned}$$

Focusing on the sum, we use the fact that $\phi_k(\alpha, y) \ll k!(\log x)^k u^{1-k}$ (see [7]) and substitute our bound for $|\alpha - s|$ to show this sum is bounded by a constant times

$$\sum_{k=1}^{\infty} \frac{(k+2) \log x}{(\log y)^{k-1}} = O(\log x).$$

We now have

$$\phi_1(s, y) = \phi_1(\alpha, y) + (s - \alpha)(\phi_2(\alpha, y) + O(\log x)).$$

Using Theorem 4 and the lower bound $\phi_2(\alpha, y) \gg (\log x)^2/u$ (see [7]) completes the proof. \square

Corollary 7. *Under the same hypotheses as the previous theorems, we have*

$$HT_f(x, y, z, \alpha_f) = HT(x, y, \alpha) \left(1 + O\left(\frac{1}{u} + \frac{1}{\log y}\right) \right).$$

Furthermore, Algorithm HT-fast has a running time of

$$O(\sqrt{y}(1/\log \log y + (\log \log x)/\log y))$$

floating point operations.

Proof. Our results above together with our choice for z and the results in [9] completes the proof.

4 Experimental Results

In this section we conclude with a comparison between algorithms HT and HT-fast.

We implemented both algorithms in C++ and had them compute a list of estimated values for $\Psi(x, y)$ with x ranging from 2^{25} up to 2^{1000} and $y = 2^{16}, 2^{18}, 2^{20}, 2^{22}$. Due to memory restrictions, it is difficult for algorithm HT to handle y values much larger than 2^{22} .

For each estimate of $\Psi(x, y)$, we also give the values of α , ζ , ϕ_1 , and ϕ_2 computed by each algorithm. We also give the elapsed time in CPU seconds (we used a Pentium Pro 200 running Linux kernel 2.2.7). Finally, we give the ratio HT_f/HT which compares the two estimates of $\Psi(x, y)$.

We used a separate table for each y -value, with the x -values indicated along the left.

Although our theory sets the condition that $5 \log x < \sqrt{y}$, in practice we used the somewhat looser condition $\log x < \sqrt{y}$ with no ill effects.

Table 1. Experimental Results: $y = 2^{16}$

x	Algorithm	α	$\zeta(\alpha, y)$	$\phi_1(\alpha, y)$	$\phi_2(\alpha, y)$	$\Psi(x, y)$	Time HT_f/HT
2^{25}	HT	0.919503	58.92	-17.33	113.8	1.992e+07	0.2
	HT-fast	0.919983	59.29	-17.38	114.3	2.004e+07	0.01 1.0056
2^{50}	HT	0.820435	689.5	-34.66	256.6	4.672e+13	0.19
	HT-fast	0.820947	699.9	-34.79	257.8	4.739e+13	0.01 1.0143
2^{75}	HT	0.767136	6700	-51.99	405.6	3.613e+19	0.2
	HT-fast	0.767657	6863	-52.2	407.5	3.698e+19	0.01 1.0234
2^{100}	HT	0.730847	5.952e+04	-69.31	557.9	1.378e+25	0.2
	HT-fast	0.731373	6.153e+04	-69.61	560.6	1.423e+25	0.01 1.0327
2^{250}	HT	0.621485	1.36e+10	-173.3	1504	1.33e+55	0.18
	HT-fast	0.622015	1.486e+10	-174.1	1512	1.451e+55	0.01 1.0911

Table 2. Experimental Results: $y = 2^{18}$

x	Algorithm	α	$\zeta(\alpha, y)$	$\phi_1(\alpha, y)$	$\phi_2(\alpha, y)$	$\Psi(x, y)$	Time HT_f/HT
2^{25}	HT	0.945145	48.97	-17.33	124.3	2.405e+07	0.74
	HT-fast	0.945437	49.17	-17.36	124.6	2.414e+07	0.01 1.0035
2^{50}	HT	0.854856	460.3	-34.66	282.9	9.399e+13	0.73
	HT-fast	0.855158	464.5	-34.74	283.7	9.481e+13	0.01 1.0088
2^{75}	HT	0.806604	3605	-51.99	448.9	1.368e+20	0.73
	HT-fast	0.806906	3657	-52.12	450.2	1.387e+20	0.01 1.0141
2^{100}	HT	0.773847	2.589e+04	-69.31	618.7	1.059e+26	0.73
	HT-fast	0.774149	2.64e+04	-69.5	620.5	1.079e+26	0.01 1.0194
2^{250}	HT	0.675444	1.713e+09	-173.3	1674	1.68e+58	0.65
	HT-fast	0.675737	1.802e+09	-173.8	1679	1.767e+58	0.02 1.0514
2^{500}	HT	0.605253	6.865e+16	-346.6	3491	9.633e+105	0.65
	HT-fast	0.605536	7.591e+16	-347.6	3501	1.065e+106	0.01 1.1052

Table 3. Experimental Results: $y = 2^{20}$

x	Algorithm	α	$\zeta(\alpha, y)$	$\phi_1(\alpha, y)$	$\phi_2(\alpha, y)$	$\Psi(x, y)$	Time HT_f/HT
2^{25}	HT	0.964492	42.35	-17.33	134.1	2.743e+07	2.61
	HT-fast	0.964689	42.47	-17.36	134.4	2.75e+07	0.02 1.0025
2^{50}	HT	0.881284	333.5	-34.66	308.4	1.581e+14	2.61
	HT-fast	0.881481	335.6	-34.72	309	1.591e+14	0.02 1.006
2^{75}	HT	0.837112	2195	-51.99	491.3	3.745e+20	2.62
	HT-fast	0.837304	2216	-52.08	492.2	3.781e+20	0.03 1.0094
2^{100}	HT	0.807214	1.327e+04	-69.31	678.6	5.018e+26	2.62
	HT-fast	0.807403	1.344e+04	-69.44	679.9	5.081e+26	0.03 1.0127
2^{250}	HT	0.717711	3.206e+08	-173.3	1845	4.277e+60	2.62
	HT-fast	0.717887	3.309e+08	-173.6	1848	4.414e+60	0.03 1.032
2^{500}	HT	0.654054	2.514e+15	-346.6	3852	6.884e+111	2.31
	HT-fast	0.654219	2.673e+15	-347.2	3859	7.316e+111	0.02 1.0628
2^{1000}	HT	0.592735	4.912e+28	-693.1	7957	9.998e+204	2.62
	HT-fast	0.592889	5.516e+28	-694.4	7970	1.123e+205	0.03 1.1228

Table 4. Experimental Results: $y = 2^{22}$

x	Algorithm	α	$\zeta(\alpha, y)$	$\phi_1(\alpha, y)$	$\phi_2(\alpha, y)$	$\Psi(x, y)$	Time	HT_f/HT
2^{25}	HT	0.97945	37.7	-17.33	143.5	3.013e+07	9.39	
	HT-fast	0.979551	37.75	-17.34	143.6	3.016e+07	0.05	1.0013
2^{50}	HT	0.902068	256.7	-34.66	333.2	2.351e+14	9.56	
	HT-fast	0.902169	257.5	-34.69	333.5	2.358e+14	0.04	1.003
2^{75}	HT	0.861256	1463	-51.99	532.7	8.177e+20	9.54	
	HT-fast	0.861354	1470	-52.04	533.3	8.216e+20	0.05	1.0048
2^{100}	HT	0.833713	7676	-69.31	737.4	1.692e+27	9.51	
	HT-fast	0.833809	7726	-69.39	738.2	1.703e+27	0.05	1.0065
2^{250}	HT	0.75155	8.081e+07	-173.3	2014	3.469e+62	9.56	
	HT-fast	0.751638	8.212e+07	-173.5	2016	3.525e+62	0.05	1.0162
2^{500}	HT	0.693294	1.645e+14	-346.6	4213	3.274e+116	9.5	
	HT-fast	0.693376	1.697e+14	-346.9	4217	3.376e+116	0.05	1.0312
2^{1000}	HT	0.637262	2.293e+26	-693.1	8712	1.051e+216	9.53	
	HT-fast	0.637338	2.43e+26	-693.8	8720	1.114e+216	0.05	1.0597

5 Conclusions

In this paper we have shown how to drastically speed algorithm HT for computing estimates of the function $\Psi(x, y)$. Our new algorithm appears to be quite accurate, and it is much faster than algorithm HT for larger values of y .

Acknowledgements

The author wishes to thank Walter Gautschi for his help. Thanks also to the Department of Computer Sciences at Purdue University for hosting the author during his Fall 1998 sabbatical.

The National Science Foundation provided support under NSF Grant CCR-9626877.

Finally, former Butler students Rachel Butler and Tina Hobbs began preliminary work on this project in the summer of 1998 as part of the Butler Summer Institute. In particular, Rachel gave estimates for ϕ_1 and ϕ_2 that, though inferior to what is presented here, were original just the same.

References

1. M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, 1970.
2. E. Bach and J. Shallit. *Algorithmic Number Theory*, volume 1. MIT Press, 1996.
3. Daniel J. Bernstein. Enumerating and counting smooth integers. Chapter 2, PhD Thesis, University of California at Berkeley, May 1995.
4. Daniel J. Bernstein. Bounding smooth integers. In J. P. Buhler, editor, *Proceedings of the Third International Algorithmic Number Theory Symposium*, pages 128–130, Portland, Oregon, June 1998. Springer. LNCS 1423.

5. E. R. Canfield, P. Erdős, and C. Pomerance. On a problem of Oppenheim concerning “Factorisatio Numerorum”. *Journal of Number Theory*, 17:1–28, 1983.
6. A. Hildebrand. On the number of positive integers $\leq x$ and free of prime factors $> y$. *Journal of Number Theory*, 22:289–307, 1986.
7. A. Hildebrand and G. Tenenbaum. On integers free of large prime factors. *Trans. AMS*, 296(1):265–290, 1986.
8. A. Hildebrand and G. Tenenbaum. Integers without large prime factors. *Journal de Théorie des Nombres de Bordeaux*, 5:411–484, 1993.
9. S. Hunter and J. P. Sorenson. Approximating the number of integers free of large prime factors. *Mathematics of Computation*, 66(220):1729–1741, 1997.
10. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.
11. Pieter Moree. *Psixyology and Diophantine Equations*. PhD thesis, Rijksuniversiteit Leiden, 1993.
12. Karl K. Norton. *Numbers with Small Prime Factors, and the Least k th Power Non-Residue*, volume 106 of *Memoirs of the American Mathematical Society*. American Mathematical Society, Providence, Rhode Island, 1971.
13. C. Pomerance, editor. *Cryptology and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society, Providence, Rhode Island, 1990.
14. J. P. Sorenson. Trading time for space in prime number sieves. In J. Buhler, editor, *Proceedings of the Third International Algorithmic Number Theory Symposium*, pages 179–195, Portland, Oregon, June 1998. Springer.

Computing All Integer Solutions of a General Elliptic Equation

Roel J. Stroeker¹ and Nikolaos Tzanakis²

¹ Econometric Institute - Erasmus University
3000 DR Rotterdam, The Netherlands
`stroeker@few.eur.nl`

`http://www.few.eur.nl/few/people/stroeker/`
² Department of Mathematics, University of Crete

GR-71409 Iraklion, Crete, Greece
`tzanakis@math.uch.gr`

`http://www.math.uch.gr/selidamath/faculty/tzanakis.htm`

Abstract. The *Elliptic Logarithm Method* has been applied with great success to the problem of computing all integer solutions of equations of degree 3 and 4 defining elliptic curves. We explore the possibility of extending this method to include any equation $f(u, v) = 0$, where $f \in \mathbb{Z}[u, v]$ defines an irreducible curve of genus 1, independent of shape or degree of the polynomial f . We give a detailed description of the general features of our approach, putting forward along the way some claims (one of which conjectural) that are supported by the explicit examples added at the end.

1 Introduction

Throughout this paper, the term *elliptic equation* shall mean an equation $f(u, v) = 0$ in rational integers u and v , where $f \in \mathbb{Z}[X, Y]$ is such that the plane curve defined by $f = 0$ is an irreducible curve of genus 1. The *Elliptic Logarithm Method*—`Elllog` for short—as a *practical* method for solving such equations, was first applied by Stroeker and Tzanakis [12] and, independently, by Gebel, Pethő and Zimmer [6]. Since then, it has been applied extensively to a variety of elliptic equations of degree 3 or 4; see [11], [16], [1], [7], [14], [15], [13]. In particular, a general treatment of the cubic elliptic equation can be found in [15].

Now that many equations have been successfully solved by application of `Elllog`, it seems natural to ask what we can learn from the experience acquired so far, so that we may distinguish the essential characteristics of the method which would make its successful application possible to any elliptic equation. We shall put forward some plausible suggestions, not all of which we can prove yet in full generality. Next we shall test our general observations by a few specific examples of non-standard elliptic equations.

2 Preliminaries

Let

$$f(u, v) = 0, \text{ where } f \in \mathbb{Z}[u, v] \text{ is irreducible,}$$

define an elliptic curve \mathcal{C} , birationally equivalent over a number field \mathbb{K} of degree at most $\min\{\deg_u f, \deg_v f\}$ to

$$\mathcal{E} : y^2 = q(x) = x^3 + Ax + B,$$

by means of a birational transformation

$$\begin{aligned} u &= \mathcal{U}(x, y), v = \mathcal{V}(x, y) \\ x &= \mathcal{X}(u, v), y = \mathcal{Y}(u, v) \end{aligned}$$

(see e.g. [9], Proposition 1).

Claim 1 *One can explicitly calculate a possibly large positive constant M , and finitely many parametrizations of \mathcal{C} of the form*

$$u(t) = t^{-\nu}, \quad v(t) = \alpha t^\mu + \alpha' t^{\mu'} + \alpha'' t^{\mu''} + \dots \quad (1)$$

for rational integers $\nu \geq 1$, $\mu < \mu' < \mu'' < \dots$, and non-zero algebraic integers $\alpha, \alpha', \alpha'' \dots$, such that every real point (u, v) on \mathcal{C} with $|u| > M$ can be expressed as $(u(t), v(t))$ by means of one of the parametrizations (1) for a suitable value of t .

Although this claim seems quite classical (Puiseux), the crux lies in the effectiveness of the calculation of M . For a proof, see Lemma 5 of [2]. This result of Coates, however, is not useful for explicit computations, as it implies an extremely large M . Much smaller M is implied by subsequent results of W.M. Schmidt [8], and B.M. Dwork and A. van der Poorten [4],[5]. In certain examples the numerical values of M generated by these improved results may still be very large. For instance, the size of M in our example of section 6.3 is roughly 10^{60} ; this means we need a method that can detect, in some subtle way, all integral solutions (u, v) with $|u| < M$. At present, no such method is known to us.

Clearly, there is no loss of generality restricting our investigations to those solutions (u, v) of $f(u, v) = 0$ with $u > 0$. The above claim implies that, for a given point (u, v) on the curve and u sufficiently large, the equation $f(u, v) = 0$ can be solved for v , i.e. there exist differentiable functions $v_1(u), \dots, v_k(u)$, ($k \leq \deg_v f$), such that $f(u, v_i(u)) = 0$ identically in u for every $i \in \{1, \dots, k\}$. Of course, this is also an immediate consequence of the *Implicit Function Theorem*. Let us put

$$x_{0i} = \lim_{u \rightarrow \infty} \mathcal{X}(u, v_i(u)).$$

From this point onwards $P \in \mathcal{C}$ will always denote a point with integral coordinates $(u(P), v(P))$. Since all points P with relatively small coordinates

can be easily found explicitly, we may assume $u(P)$ to be sufficiently large, so that, for some $i \in \{1, \dots, k\}$, $v(P) = v_i(u(P))$ and $x(P) = \mathcal{X}(u(P), v(P))$ is close to x_{0i} . Let us explain what we mean by ‘close’ in this context.

Let e_1 denote the only real root of $q(x) = 0$ if this equation has a single root only (the complex case), or the largest real root in case of three real roots (the real case); in the latter case the other two real roots are denoted by e_2 and e_3 and we assume $e_3 < e_2 < e_1$. In the complex case, $x_{0i} \geq e_1$ and by ‘close’ we mean that $x(P) \geq e_1$ as well. In the real case, $x_{0i} \in [e_3, e_2]$ and now ‘close’ means that $x(P) \in [e_3, e_2]$ too.

3 Two Related Elliptic Integrals

It is not difficult to see that

$$\frac{dx}{y} = G(u, v) \frac{du}{f_v(u, v)}, \quad (2)$$

where

$$G(u, v) = 2 \frac{\mathcal{Y}_u(u, v) \cdot f_v(u, v) - \mathcal{Y}_v(u, v) \cdot f_u(u, v)}{3\mathcal{X}^2(u, v) + A}.$$

In case $f(u, v) = 0$ is a Weierstrass equation, a quartic equation of type $v^2 = Q(u)$ for some quartic polynomial Q , or a general cubic elliptic equation, the function $G(u, v) \in \mathbb{C}(\mathcal{C})$ is constant; see [12], [16] and [15]. For example, in case of a general cubic equation, $G(u, v) = \pm 2$.

Now fix $i \in \{1, \dots, k\}$. For u sufficiently large, $\mathcal{Y}(u, v_i(u))$ and $\mathcal{X}(u, v_i(u))$ are continuous functions of u ; if we denote them by $y(u)$ and $x(u)$ respectively, then $y(u)^2 = x(u)^3 + Ax(u) + B = q(x(u))$. Hence $y(u) = \varepsilon\sqrt{q(x(u))}$ with $\varepsilon \in \{-1, 1\}$. On putting

$$g_i(u) = G(u, v_i(u)),$$

we have, by (2) and our assumption on the size of $u(P)$,

$$\int_{u(P)}^{\infty} \frac{g_i(u) du}{f_v(u, v_i(u))} = \int_{x(P)}^{x_{0i}} \frac{dx}{\varepsilon\sqrt{q(x)}}. \quad (3)$$

Here $x(P) = \mathcal{X}(u(P), v(P))$ of course.

4 Necessary Conditions for the Applicability of **Ellog**

For **Ellog** to work it is essential that the integral in the left-hand side of (3) tends to zero as $u(P)$ tends to ∞ .

Conjectural Claim 2

$$\frac{g_i(u)}{f_v(u, v_i(u))} = \mathcal{O}(u^{-1-\delta}) \quad (4)$$

for some $\delta > 0$.

For example, if $f(u, v) = 0$ happens to be a Weierstrass equation to start with, no birational transformation is needed, and $\delta = \frac{1}{2}$, while in case of either a non-Weierstrass cubic equation or of a quartic equation of type $v^2 = Q(u)$ with quartic polynomial Q , it is easily shown that $\delta = 1$ (see [15] and [16], respectively).

It follows from (4) that the left-hand side of (3) is $\leq c_1 u^{-\delta}$. Here the constant c_1 , as well as all other constants c_i in the sequel are effectively computable.

Claim 3 *Let $h(\cdot)$ denote the logarithmic height. Then,*

$$h(x(P)) = h(\mathcal{X}(u(P), v(P))) \leq c_2 + c_3 \log |u(P)|. \quad (5)$$

Inequality (5) is easily seen to be true. Indeed, write

$$f(u, v) = f_d(u)v^d + \dots + f_1(u)v + f_0(u)$$

with $f_j(u) \in \mathbb{Z}[u]$ of degree j . If $(u, v) \in \mathbb{Z}^2$ and $f(u, v) = 0$, then v is an integral root of the polynomial $f_d(u)X^d + \dots + f_1(u)X + f_0(u)$ with integer coefficients. Hence v divides $f_0(u)$, from which it follows that $|v| \leq |f_0(u)|$. This, combined with the fact that $\mathcal{X}(u, v)$ is a rational function of u and v with integer coefficients, implies inequality (5).

We also need the following relation between the Néron-Tate height and the logarithmic height (see e.g. [10]):

$$\hat{h}(x(P)) - \frac{1}{2}h(P) \leq c_4. \quad (6)$$

Now, the right-hand side of (3) is a so-called linear form in elliptic logarithms of points on $\mathcal{E}(\overline{\mathbb{Q}})$, say $\mathcal{L}(P)$. It has integer coefficients, which are essentially the coefficients of P with respect to a Mordell-Weil basis chosen well in advance, and we denote the maximum absolute value of these coefficients by N . A more detailed description of \mathcal{L} is given in section 5.

By S. David's Theorem [3], we obtain a lower bound for $\mathcal{L}(P)$ of the shape

$$|\mathcal{L}(P)| > \exp(-c_5(\log N + c_6)(\log \log N + c_7)^k), \quad (7)$$

where $k = r + 2$ or $r + 3$ and r is the rank of the Mordell-Weil group. We also need an upper bound for $\mathcal{L}(P)$. This upper bound can be deduced from (3) and (4):

$$|\mathcal{L}(P)| \leq c_1(u(P))^{-\delta}.$$

Combining this with (5), (6) and the well-known fact that $\hat{h}(P) \geq c_8 N^2$, we obtain

$$|\mathcal{L}(P)| \leq \exp(-c_9 N^2 + c_{10}) \quad (8)$$

and finally (7) and (8) imply an upper bound for N . Much of the material found in this section and the next consists of straightforward adaptations from [12], [16] or [15].

5 The Linear Form $\mathcal{L}(P)$

In this section we discuss in some detail the linear form $\mathcal{L}(P)$, and we show that this form indeed qualifies as a suitable linear form in elliptic logarithms of points on $\mathcal{E}(\overline{\mathbb{Q}})$ to which S. David's theorem, mentioned in the previous section, can be applied.

The curve $\mathcal{E}(\mathbb{R})$, defined by $y^2 = q(x)$, has the identity component $\mathcal{E}_0(\mathbb{R})$ and in the real case—we remind the reader that $q(x) = 0$ then has three real roots $e_1 > e_2 > e_3$ —also the bounded component $\mathcal{E}_1(\mathbb{R})$. Let $Q_j = (e_j, 0) \in \mathcal{E}(\overline{\mathbb{Q}})$ for $j = 1, 2, 3$. For any $R \in \mathcal{E}_1(\mathbb{R})$ we put $R' = R + Q_2 \in \mathcal{E}_0(\mathbb{R})$. We have the usual isomorphism

$$\phi : \mathcal{E}_0(\mathbb{R}) \longrightarrow [0, 1] = \mathbb{R}/\mathbb{Z}$$

(see e.g. [12]). In the complex case—that is when $q(x) = 0$ has a single real root— $\mathcal{E}_0(\mathbb{R}) = \mathcal{E}(\mathbb{R})$ and ϕ is defined on the whole of $\mathcal{E}(\mathbb{R})$. In the real case ϕ is extended to a two-to-one epimorphism $\tilde{\phi}$, defined as follows:

$$\tilde{\phi}(R) = \begin{cases} \phi(R) & \text{if } R \in \mathcal{E}_0(\mathbb{R}), \\ \phi(R') & \text{if } R \in \mathcal{E}_1(\mathbb{R}). \end{cases}$$

Let $\omega = 2 \int_{e_1}^{\infty} \frac{dt}{\sqrt{q(t)}}$, the fundamental real period. A bit of thought suffices to convince one that

$$\omega \cdot \tilde{\phi}(R) = \begin{cases} \text{elliptic log of } R & \text{if } R \in \mathcal{E}_0(\mathbb{R}), \\ \text{elliptic log of } R' & \text{if } R \in \mathcal{E}_1(\mathbb{R}). \end{cases} \quad (9)$$

We write

$$P = n_1 P_1 + \cdots + n_r P_r + T,$$

where P_1, \dots, P_r form a Mordell-Weil basis and T is one of the finitely many torsion points. It is easy to see that the $\tilde{\phi}(T)$ are rational numbers with effectively bounded denominators. Then,

$$\tilde{\phi}(P) \text{ and } \tilde{\phi}(-P) \text{ are of the form } m_1 \tilde{\phi}(P_1) + \cdots + m_r \tilde{\phi}(P_r) + m_0 + \frac{s}{t}, \quad (10)$$

where $m_j = \pm n_j$ ($j = 1, \dots, r$), $m_0 \in \mathbb{Z}$ is effectively bounded in terms of N , and s, t are relatively prime integers, effectively bounded by a small number.

Consider the integral in the right-hand side of (3) and recall that $f(u, v_i(u)) = 0$, provided u is sufficiently large.

Claim 4

$$x_{0i} \in \overline{\mathbb{Q}} \cup \{\pm\infty\}.$$

The truth of this statement depends only on the truth of Claim 1 as we shall see shortly. First note that $f(u, v)$ cannot be a factor of either the numerator or the denominator of the rational function $\mathcal{X}(u, v)$. For, otherwise, the whole curve \mathcal{C} could be mapped into a line, which is impossible for a curve of genus 1. Next,

by Claim 1, every point $(u, v_i(u)) \in \mathcal{C}$ with u near ∞ has a parametrization (1), where the coefficients and the exponents depend solely on the function v_i . On substituting the t -expressions for u and v of (1), the value of $\mathcal{X}(u, v_i(u))$ for u near ∞ can be seen to be given by an expression of the form

$$\frac{\beta t^\lambda + \beta' t^{\lambda'} + \beta'' t^{\lambda''} + \dots}{\gamma t^\rho + \gamma' t^{\rho'} + \gamma'' t^{\rho''} + \dots} \quad (t \text{ near zero}),$$

where $\beta, \beta', \beta'', \dots, \gamma, \gamma', \gamma'', \dots$ are non-zero algebraic numbers and $\lambda < \lambda' < \lambda'' < \dots$ and $\rho < \rho' < \rho'' < \dots$ are rational integers. This shows that

$$x_{0i} = \lim_{u \rightarrow \infty} \mathcal{X}(u, v_i(u)) = \begin{cases} \beta/\gamma & \text{if } \lambda = \rho, \\ \infty & \text{if } \lambda > \rho, \\ -\infty & \text{if } \lambda < \rho. \end{cases}$$

If $x_{0i} \neq \pm\infty$ we denote by Q_{0i} the point with x -coordinate x_{0i} and non-negative y -coordinate. If $x_{0i} = \pm\infty$ we set $Q_{0i} = \mathcal{O}$, the group identity.

We distinguish two cases:

1. $e_1 \leq x_{0i}$. Then, because $u(P)$ is assumed to be sufficiently large, we have $e_1 < x(P) = \mathcal{X}(u(P), v(P))$ and hence

$$\begin{aligned} \int_{x(P)}^{x_{0i}} \frac{dx}{\sqrt{q(x)}} &= \int_{x(P)}^{\infty} \frac{dx}{\sqrt{q(x)}} - \int_{x_{0i}}^{\infty} \frac{dx}{\sqrt{q(x)}} \\ &= \omega\phi(\sigma P) - \omega\phi(Q_{0i}) = \omega\tilde{\phi}(\sigma P) - \omega\tilde{\phi}(Q_{0i}). \end{aligned}$$

Here $\sigma = 1$ or -1 , depending on whether $y(P) = \mathcal{Y}(u(P), v(P))$ is non-negative or negative, respectively. This, combined with (10) and (9) shows that the integral in the right-hand side of (3) is equal to a linear form in elliptic logarithms

$$-\omega\tilde{\phi}(Q_{0i}) + (m_0 + \frac{s}{t})\omega + m_1\omega\tilde{\phi}(P_1) + \dots + m_r\omega\tilde{\phi}(P_r), \quad (11)$$

and all points appearing in it have algebraic coordinates.

2. $x_{0i} \in [e_3, e_2]$. Then, because $u(P)$ is sufficiently large, $x(P) \in (e_3, e_2)$ and

$$\begin{aligned} \int_{x(P)}^{x_{0i}} \frac{dx}{\sqrt{q(x)}} &= \int_{x(P)}^{e_2} \frac{dx}{\sqrt{q(x)}} - \int_{x_{0i}}^{e_2} \frac{dx}{\sqrt{q(x)}} = \int_{x(P')}^{\infty} \frac{dx}{\sqrt{q(x)}} - \int_{x(Q'_{0i})}^{\infty} \frac{dx}{\sqrt{q(x)}} \\ &= \omega\phi(\sigma P') - \omega\phi(Q'_{0i}) = \omega\tilde{\phi}(\sigma P) - \omega\tilde{\phi}(Q_{0i}) \end{aligned}$$

and we arrive at the same conclusion (11) as before.

6 Examples

It is not easy to find in the literature non-trivial examples of irreducible curves of genus 1 of an unusual shape, that is given by equations of degree at least

5. Therefore, with the exception of the third example, we have generated a few examples by ourselves. Further, we shall only discuss solutions (u, v) with $u > 0$ and sufficiently large.

We have chosen not to take `Elllog` ‘all the way’, for the simple reason that, once we have checked the various claims —and this is what we actually do below, except for the values of the various M ’s¹—completing the computations is merely a routine matter, be it a tedious one.

We have implemented in Maple a procedure for computing parametrizations (1), using Newton polygons (see e.g. [17]).

6.1 Three Simple Examples

We have grouped the following three equations because of their similarity; each provides a straightforward example of an elliptic equation of unusual form. In the table below we have gathered the relevant information.

Three simple elliptic equations $f(u, v) = 0$

$f(u, v)$	$u^5 + u^4 - 2v^3$	$u^6 + u^3 - 2v^2$	$u^7 + u^4 - 2v^2$
Singular points [u, v] (multiplicity)	$[0, 0](3), \infty(2)$	$[0, 0](2), \infty(4)$	$[0, 0](2), \infty(5)$
Rank r	0	1	1
Weierstrass A, B	0, 1	0, 8	0, 8
<u>Birational transformation</u>			
$\mathcal{X}(u, v)$	$-2\frac{v}{u(u+1)}$	$2\frac{2u^3+u^2+u+4v}{u(u-1)^2}$	$2\frac{u^4+u^3+2u^2+4v}{u^2(u-1)^2}$
$\mathcal{Y}(u, v)$	$\frac{u-1}{u+1}$	$4\frac{4u^4+3u^2+u+5uv+3v}{u(u-1)^3}$	$4\frac{u^5+3u^4+4u^2+3uv+5v}{u^2(u-1)^3}$
<u>Claim 1</u>			
ν	3	1	2
μ, μ', μ'', \dots	$-5, -2, 1, 4, 7, \dots$	$-3, 0, 3, 6, 9, \dots$	$-7, -1, 5, 11, 17, \dots$
$\alpha, \alpha', \alpha'', \dots$	$\rho, \frac{\rho}{3}, -\frac{\rho}{9}, \frac{5\rho}{81}, -\frac{10\rho}{243}, \dots$	$\rho, \frac{\rho}{2}, -\frac{\rho}{8}, \frac{\rho}{16}, -\frac{5\rho}{128}, \dots$	$\rho, \frac{\rho}{2}, -\frac{\rho}{8}, \frac{\rho}{16}, -\frac{5\rho}{128}, \dots$
ρ	$1/\sqrt[3]{2}$	$\pm 1/\sqrt{2}$	$\pm 1/\sqrt{2}$
k	1	2	2
<u>Conjectural Claim 2</u>			
δ	1/3	1	1/2
<u>Claim 4</u>			
$x_{i0}(u \rightarrow \infty)$	0	$4 \pm 4\sqrt{2}$	2

¹ We actually believe that the various series $v_i(t)$ do converge for $|t|$ less than some number of the order 0.1 say, but we cannot prove this.

6.2 A Parametric Family of Degree 5 Curves

In the course of constructing suitable examples, we struck on the following parametric family of elliptic equations:

$$f(u, v) = v^2(v - u - 1)(v + (2\tau - 1)u - 1) + \tau u^2(v^3 - 1) = 0. \quad (12)$$

For each value of the parameter $\tau \neq 0$, $\tau \in \mathbb{Z}$, this equation represents an elliptic curve \mathcal{C}_τ . The singular points of \mathcal{C}_τ are $(u, v) = (0, 0)$ and $(0, 1)$, both of multiplicity 2, and the point at infinity is a singular point of multiplicity 3. The birational equivalent curve \mathcal{E}_τ is

$$y^2 = x^3 + A_\tau x + B_\tau, \quad \text{with } A_\tau = -\frac{1}{3}\tau^4 \text{ and } B_\tau = \frac{2}{27}\tau^6 + \tau^3,$$

and the corresponding birational transformations are (one way only) given by

$$\mathcal{X}(u, v) = \frac{1}{3}\tau^2 - \tau v, \quad \mathcal{Y}(u, v) = \frac{\tau v(-1 + \tau u - u + v)}{u}.$$

In this example $k = 2$, i.e. there exist two parametrizations near $u = \infty$. The first parametrization is given by²

$$\begin{aligned} u_1(t) &= t^{-1}, \\ v_1(t) &= -\tau t^{-2} - 2(\tau - 1)t^{-1} + \frac{1}{\tau} + 2\frac{(\tau - 1)^2}{\tau^2}t - \frac{(4\tau - 5)(\tau - 1)^2}{\tau^3}t^2 \\ &\quad + 2\frac{(4\tau - 7)(\tau - 1)^3}{\tau^4}t^3 - \frac{16\tau^5 - 104\tau^4 + 259\tau^3 - 310\tau^2 + 182\tau - 42}{\tau^5}t^4 \\ &\quad + 2\frac{(\tau - 1)(16\tau^5 - 120\tau^4 + 333\tau^3 - 430\tau^2 + 270\tau - 66)}{\tau^6}t^5 \\ &\quad - \frac{64\tau^7 - 688\tau^6 + 2928\tau^5 - 6495\tau^4 + 8288\tau^3 - 6174\tau^2 + 2508\tau - 429}{\tau^7}t^6 \\ &\quad + O(t^7) \quad (t \rightarrow 0). \end{aligned}$$

It is obvious from this that

$$x_{10} = \lim_{u \rightarrow \infty} \mathcal{X}(u, v_1(u)) = \infty.$$

For this parametrization we find

$$\begin{aligned} \frac{g_1(u)}{f_v(u, v_1(u))} &= \frac{2}{\tau}u^{-2} - 4\frac{\tau - 1}{\tau^2}u^{-3} + 2\frac{4\tau^2 - 9\tau + 6}{\tau^3}u^{-4} \\ &\quad - \frac{8}{3}\frac{6\tau^3 - 16\tau^2 + 17\tau - 7}{\tau^4}u^{-5} + O(u^{-6}) \quad (u \rightarrow \infty), \end{aligned}$$

² Although not really necessary, we calculated quite a number of terms in order to see what they are like and to demonstrate Maple's capabilities.

so that $\delta = 1$ in this case.

The second parametrization is

$$\begin{aligned} u_2(t) &= t^{-1}, \\ v_2(t) &= \rho_\tau - \frac{2(\tau-1)(4\tau^2\rho_\tau^2 - 10\tau\rho_\tau^2 + 4\rho_\tau^2 + 17\tau^2\rho_\tau + 2\rho_\tau - 8\tau\rho_\tau - 6\tau + 3\tau^2)}{59\tau^3 - 48\tau^2 + 24\tau - 4}t \\ &\quad + \frac{1}{\tau(59\tau^3 - 48\tau^2 + 24\tau - 4)^2}(-80\rho_\tau^2 + 864\tau\rho_\tau^2 + 10328\tau^3\rho_\tau^2 + 16574\tau^5\rho_\tau^2 \\ &\quad - 17000\tau^4\rho_\tau^2 - 3904\tau^2\rho_\tau^2 - 8711\tau^6\rho_\tau^2 + 1588\tau^7\rho_\tau^2 \\ &\quad + 2088\tau^7\rho_\tau - 3458\tau^4\rho_\tau + 6074\tau^5\rho_\tau + 1192\tau^3\rho_\tau \\ &\quad - 5270\tau^6\rho_\tau - 208\tau^2\rho_\tau + 16\tau\rho_\tau + 1132\tau^7 + 80\tau \\ &\quad + 7588\tau^5 + 2808\tau^3 - 4695\tau^6 - 752\tau^2 - 6192\tau^4)t^2 \\ &\quad + O(t^3) \quad (t \rightarrow 0), \end{aligned}$$

where ρ_τ satisfies the cubic equation $X^3 + (1/\tau - 2)X^2 - 1 = 0$. For this parametrization we find

$$x_{20} = \lim_{u \rightarrow \infty} \mathcal{X}(u, v_2(u)) = \frac{1}{3}\tau^2 - \rho_\tau\tau.$$

and

$$\frac{g_2(u)}{f_v(u, v_2(u))} = d_2u^{-2} + O(u^{-3}) \quad (u \rightarrow \infty),$$

where d_2 can be (and was) explicitly calculated by Maple, but is too complicated to be included here. Because $d_2 \neq 0$, $\delta = 1$ for this parametrization as well.

6.3 An Example Taken from Maple's Help Facility

The Help Topic of the Maple V Release 5.1 command **algcurves[singularities]** makes use of the following curve of rank 5:

$$\begin{aligned} f(u, v) = & 180u^5 - 207u^4v - 8v^5 - 450u^4 + 621u^3v - 128uv^3 - 35v^4 + \\ & 369u^3 - 521u^2v + 82v^3 - 100u^2 + 135uv - 19v^2 - 7u - 28v + 8 = 0. \end{aligned}$$

Singular points (all of multiplicity 2) are $(u, v) = (0, 1), (1, 0), (1, -1)$ and the two complex points $(u, v) = (i, i), (-i, -i)$. A short Weierstrass model of this curve is

$$y^2 = x^3 - \frac{62058288278602561}{805306368}x + \frac{61852994116858326481398145}{59373627899904}.$$

The corresponding birational transformations are given by

$$\mathcal{X}(u, v) = \frac{43681 \text{ Num}X(u, v)}{49152u(u^2 + 1)(u - 1)^2},$$

$$\mathcal{Y}(u, v) = \frac{9129329 \text{ Num}Y(u, v)}{524288u(u^2 + 1)(u - 1)^3},$$

with

$$\begin{aligned} \text{Num}\mathcal{X}(u, v) &= 103981u^5 + 15228u^4v + 10284u^3v^2 + 1536u^2v^3 \\ &\quad + 4128uv^4 - 316526u^4 + 47412u^3v + 67584u^2v^2 + 15468uv^3 \\ &\quad - 2592v^4 + 368606u^3 - 71388u^2v - 88968uv^2 - 13932v^3 \\ &\quad - 206150u^2 + 2268uv + 12636v^2 + 52681u + 6480v - 2592, \\ \text{Num}\mathcal{Y}(u, v) &= 2070033u^6 + 70533u^5v - 28045u^4v^2 \\ &\quad + 45962u^3v^3 + 90616u^2v^4 - 7973144u^5 + 1130670u^4v \\ &\quad + 1634455u^3v^2 + 312517u^2v^3 - 117296uv^4 + 12052790u^4 \\ &\quad - 2569492u^3v - 3224660u^2v^2 + 524456uv^3 + 33368v^4 \\ &\quad - 9090868u^3 + 1336366u^2v + 1787607uv^2 + 179353v^3 \\ &\quad + 3599145u^2 + 115343uv - 162669v^2 - 691324u - 83420v \\ &\quad + 33368. \end{aligned}$$

In this example there exists only one parametrization near $u = \infty$, given by

$$\begin{aligned} u_1(t) &= t^{-1}, \\ v_1(t) &= \rho t^{-1} + d_0(\rho) + d_1(\rho)t + d_2(\rho)t^2 + O(t^3) \quad (t \rightarrow 0) \end{aligned}$$

with

$$\begin{aligned} d_0(\rho) &= \frac{117652915}{2647875139}\rho^4 + \frac{59690773}{294208348}\rho^3 + \frac{64881275}{294208348}\rho^2 - \frac{37533284}{73552087}\rho + \frac{3292350}{73552087}, \\ d_1(\rho) &= \frac{2409249577008465}{86558552032889104}\rho^4 - \frac{143100375932054279}{4154810497578676992}\rho^3 - \frac{3841218563243545585}{12464431492736030976}\rho^2 \\ &\quad - \frac{442118719850886867}{692468416263112832}\rho + \frac{99742932488150451}{173117104065778208}, \\ d_2(\rho) &= -\frac{46304367990791457732640885}{3667139798237041673525787648}\rho^4 + \frac{91871979044861844697522343}{1833569899118520836762893824}\rho^3 \\ &\quad + \frac{43666801880702130891932691}{814919955163787038561286144}\rho^2 + \frac{2831900188941035651896208357}{29337118385896333388206301184}\rho \\ &\quad - \frac{213000092757640705570148071}{814919955163787038561286144}, \end{aligned}$$

where ρ is the only real root of $8X^5 + 207X - 180 = 0$. Standard, but tedious computations yield

$$x_{10} = \lim_{u \rightarrow \infty} \mathcal{X}(u, v_1(u)) = \frac{43681}{49152}(4128\rho^4 + 1536\rho^3 + 10284\rho^2 + 15228\rho + 103981)$$

and finally

$$\begin{aligned} \frac{g_1(u)}{f_v(u, v_1(u))} &= \left(-\frac{3208960}{19764496521}\rho^4 - \frac{3488000}{19764496521}\rho^3 + \frac{3609248}{6588165507}\rho^2 + \frac{18542144}{6588165507}\rho \right. \\ &\quad \left. - \frac{7380608}{2196055169} \right) u^{-2} + O(u^{-3}) \quad (u \rightarrow \infty), \end{aligned}$$

which in particular implies that $\delta = 1$.

References

1. Bremner, A., Stroeker, R.J., Tzanakis, N.: On sums of consecutive squares. *J. Number Th.* **62** (1997) 39–70
2. Coates, J.: Construction of rational functions on a curve. *Proc. Camb. Philos. Soc.* **68** (1970) 105–123
3. David, S.: Minorations de formes linéaires de logarithmes elliptiques. *Mémoires Soc. Math. France (N.S.)* **62** (1995)
4. Dwork, B.M., van der Poorten, A.: The Eisenstein constant. *Duke Math. J.* **65** (1992) 23–43
5. Dwork, B.M., van der Poorten, A.: Corrections to “The Eisenstein constant”. *Duke Math. J.* **76** (1994) 669–672
6. Gebel, J., Pethő, A., Zimmer, H.G.: Computing integral points on elliptic curves. *Acta Arith.* **68** (1994) 171–192
7. Gebel, J., Pethő, A., Zimmer, H.G.: On Mordell’s equation. *Compositio Math.* **110** (1998) 335–367
8. Schmidt, W.M.: Eisenstein’s theorem on power series expansions of algebraic functions. *Acta Arithm.* **56** (1990) 161–179
9. Schmidt, W.M.: Integer points on curves of genus 1. *Compositio Math.* **81** (1992) 33–59
10. Silverman, J.H.: The difference between the Weil height and the canonical height on elliptic curves. *Math. Comp.* **55** (1990) 723–743
11. Stroeker, R.J.: On the sum of consecutive cubes being a perfect square. *Compositio Math.* **97** (1995) 295–307
12. Stroeker, R.J., Tzanakis, N.: Solving elliptic diophantine equations by estimating linear forms in elliptic logarithms. *Acta Arith.* **67** (1994) 177–196
13. Stroeker, R.J., Tzanakis, N.: On the Elliptic Logarithm Method for Elliptic Diophantine Equations: Reflections and an improvement. *Experim. Math.* **8** (1999) 135–149
14. Stroeker, R.J., de Weger, B.M.M.: Elliptic Binomial Diophantine Equations. *Math. Comp.* **68** (1999) 1257–1281
15. Stroeker, R.J., de Weger, B.M.M.: Solving elliptic diophantine equations: the general cubic case. *Acta Arith.* **87** (1999) 339–365
16. Tzanakis, N.: Solving elliptic diophantine equations by estimating linear forms in elliptic logarithms. The case of quartic equations. *Acta Arith.* **75** (1996) 165–190
17. Walker, R.J.: Algebraic Curves. Springer-Verlag, New-York 1978

A Note on Shanks's Chains of Primes

Edlyn Teske¹ and Hugh C. Williams^{*2}

¹ Dept. of Combinatorics and Optimization
Centre for Applied Cryptographic Research, University of Waterloo
Waterloo, ON N2L 3G1, Canada
eteske@math.uwaterloo.ca

² Dept. of Computer Science, University of Manitoba
Winnipeg, MB R3T 2N2, Canada
williams@cs.umanitoba.ca

Abstract. For integers a and b we define the Shanks chain p_1, p_2, \dots, p_k of length k to be a sequence of k primes such that $p_{i+1} = ap_i^2 - b$ for $i = 1, 2, \dots, k-1$. While for Cunningham chains it is conjectured that arbitrarily long chains exist, this is, in general, not true for Shanks chains. In fact, with $s = ab$ we show that for all but 56 values of $s \leq 1000$ any corresponding Shanks chain must have bounded length. For this, we study certain properties of functional digraphs of quadratic functions over prime fields, both in theory and practice. We give efficient algorithms to investigate these properties and present a selection of our experimental results.

1 Introduction

Let $\epsilon \in \{+1, -1\}$ be fixed. A *Cunningham chain* $p_1, p_2, p_3, \dots, p_n$ of length k (see Guy [9], §A7) is a sequence of k primes such that

$$p_{i+1} = 2p_i + \epsilon \quad (i = 1, 2, \dots, k-1).$$

For example, if $\epsilon = 1$, we say that

$$2, 5, 11, 23, 47$$

is a Cunningham chain of length 5. The longest known chains of Cunningham primes have recently been determined by Forbes [8]. For $\epsilon = 1$ the longest chains have $k = 14$ (one of these has $p_1 = 23305436881717757909$), and for $\epsilon = -1$, the longest known chain has $k = 16$ ($p_1 = 3203000719597029781$). Indeed by Schinzel's [16] [17] Conjecture H one would expect for either value of ϵ and any given $k \geq 1$ the existence of an infinitude of Cunningham chains of length k . Consider now the quantitative version of Conjecture H, given by Bateman and Horn [4].

* Research supported by NSERC of Canada grant #A7649

Hypothesis H. Suppose f_1, f_2, \dots, f_k are polynomials in one variable with all coefficients integral and leading coefficients positive, their degrees being h_1, h_2, \dots, h_k respectively. Suppose each of these polynomials is irreducible over the field of rational numbers and no two of them are identical. Let $P(N)$ denote the number of positive integers n between 1 and N inclusive such that $f_1(n), f_2(n), \dots, f_k(n)$ are all (positive) primes. Then as $N \rightarrow +\infty$ we have

$$P(N) = \frac{C(f_1, f_2, \dots, f_k)}{h_1 h_2 \cdots h_k} \int_2^N \frac{du}{(\log u)^k} + o\left(\int_2^N \frac{du}{(\log u)^k}\right),$$

where

$$C(f_1, f_2, \dots, f_k) = \prod_q \left\{ \left(1 - \frac{1}{q}\right)^{-k} \left(1 - \frac{\omega(q)}{q}\right) \right\},$$

the product being extended over all primes and $\omega(q)$ being the number of solutions of the congruence

$$f_1(x) f_2(x) \cdots f_k(x) \equiv 0 \pmod{q}. \quad (1.1)$$

For the case of Cunningham chains of length k we have

$$f_1(x) = x, \quad f_2(x) = 2x + \epsilon, \quad f_3(x) = 4x + 3\epsilon, \dots, \quad f_k(x) = 2^{k-1}x + (2^{k-1} - 1)\epsilon.$$

For any given odd prime q , let $\nu(q)$ denote the multiplicative order of 2 modulo q . We must have $\nu(q) \mid q - 1$ and

$$\omega(q) = \begin{cases} k & \text{when } k \leq \nu(q) \\ \nu(q) & \text{otherwise.} \end{cases}$$

Thus, $\omega(q) \leq q - 1 < q$ and $C(f_1, f_2, \dots, f_k)$ is positive. We can therefore assert under this conjecture that if $P(N)$ is the number of Cunningham chains of length k starting with some $p_1 < N$, then as $N \rightarrow \infty$, we have

$$P(N) \sim C(f_1, f_2, \dots, f_k) \int_2^N \frac{du}{(\log u)^k}.$$

Evidently, $P(N)$ goes to infinity as N does.

In 1963 Shanks [19] observed (under Hardy and Littlewood's [10] Conjecture F) that if

$$Q(n) = \#\{x : 0 \leq x \leq n, x^2 - 17 \text{ is prime}\},$$

then

$$Q(n) \sim 1.1803 \int_2^n \frac{dx}{\log x}.$$

Since the value of the constant 1.1803 exceeds 1, this caused him to write the following passage in a letter [18] to D.H. and Emma Lehmer in 1969.

“ $n^2 - 17$ has a higher prime density than n itself, even though it grows twice as fast....

It follows that prime chains

$$p_{i+1} = (2p_i)^2 - 17$$

should be [a] little longer than

$$p_{i+1} = 2p_i + 1$$

even though they grow twice as fast. I never did run it though. Try it sometime.”

There is no record of any response to this by the Lehmers, possibly because the doubly exponential growth rate for the p_i values rapidly produces numbers that certainly would have been too difficult to test for primality by the methods available at the time. In a short computer trial we discovered that if $p_1 = 3$, then p_1, p_2, p_3, p_4 are all primes and if $p_1 = 303593$, then p_1, p_2, p_3, p_4, p_5 are all primes. We were unable to find a sequence of 6 primes for any $p_1 < 6200000$.

For a given pair of integers a, b , we set $f(x) = ax^2 - b$ and define $f_1(x) = x$, $f_{i+1}(x) = f(f_i(x))$ ($i = 1, 2, \dots$). We define the corresponding *Shanks chain* of length k to be a set of primes

$$p_1, p_2, p_3, \dots, p_k$$

such that $p_i = f_i(p_1)$ ($i = 1, 2, \dots, k$). It seems that Shanks believed that when $a = 4$ and $b = 17$ one might be able to get somewhat longer chains of primes (starting with $p_1 < N$ for a given N) than the Cunningham chains. However, this is not the case. For consider the prime 59. It turns out that for any integer x one finds that

$$f_i(x) \equiv 0 \pmod{59}$$

for some $i \leq 17$. Thus, the maximum chain length possible for this Shanks chain is 16. That is, if $k \geq 17$, then the number of solutions of (1.1) is 59 when $q = 59$ or $\omega(59) = 59$. Hence, $C(f_1, f_2, \dots, f_k) = 0$ if $k \geq 17$. The question that now comes to mind is: how often does this phenomenon occur? To investigate this question we first note that

$$af_{i+1}(x) = (af_i(x))^2 - ab.$$

Thus, for an integer s we will define $g(x, s) = x^2 - s$, $g_0(x, s) = x$, $g_{i+1}(x, s) = g(g_i(x, s))$ ($i = 0, 1, 2, \dots$). It follows that if $s = ab$, we get

$$f_i(x) = g_{i-1}(ax)/a, \tag{1.2}$$

and if q does not divide a , then (1.1) has just as many solutions as

$$\prod_{i=0}^{k-1} g_i(x, s) \equiv 0 \pmod{q}. \tag{1.3}$$

We now turn our attention to the problem of determining whether for a fixed integer s there exists some prime q and some minimal $\kappa(>0) = \kappa(s)$ such that q does not divide s and for any integer x

$$g_i(x, s) \equiv 0 \pmod{q} \quad (1.4)$$

for some $i \leq \kappa$. That is, when is $\omega(q) = q$ for $f_i(x)$ given by (1.2) and $k > \kappa$? We have already seen that if $s = 9$ ($\equiv 68 = 4 \cdot 17 \pmod{59}$), then this must be the case for $p = 59$ and $\kappa = 16$. In Table 1 we present for values of $q \leq 200$ those values of $s \pmod{q}$ and corresponding values of κ such that (1.4) must hold for some $i \leq \kappa$. Note that for these values of s the maximum possible chain length of a Shanks chains is $\kappa(s)$.

Table 1.

q	$s, \kappa(s)$	q	$s, \kappa(s)$	q	$s, \kappa(s)$
2		59	9,17; 25,11	137	87,27; 118,31
3	1,2	61	39,15; 45,11; 48,9	139	41,22; 107,21
5	4,3	67	62,16	149	96,32; 129,24
7	1,3; 4,3	71		151	94,35; 127,20; 137,27
11	5,5	73	61,22;	157	16,33; 100,32
13	9,6; 10,5	79	8,17	163	84,20; 135,29
17		83	37,21; 51,21	167	49,26
19	9,7; 17,7	89	10,18	173	151,40
23	1,9	97	54,18; 66,25	179	
29	5,8; 22,10; 25,11	101	54,25	181	136,25
31	16,8; 19,10	103	18,20	191	64,24; 104,24
37	33,13	107	102,22	193	4,39; 126,25
41	25,15; 39,13; 40,9	109	38,17; 82,27	197	
43	17,8	113		199	47,26; 98,32; 103,23
47		127	35,20; 74,30; 87,17		
53	44,14	131			

A glance at Table 1 reveals that for many values of s we would not expect to have arbitrarily long Shanks chains. The purpose of this paper is to examine when $\omega(q) = q$ for a particular value of s .

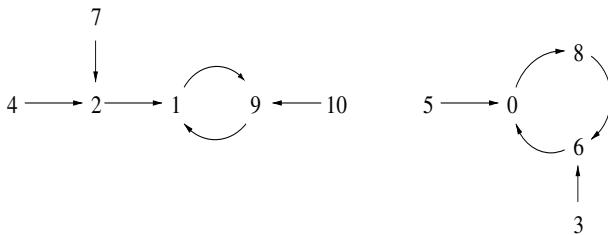
2 Generators

Clearly, we are dealing here with a problem which involves iterating a nonlinear function modulo a prime; such problems, even for functions as simple as quadratic polynomials, are notoriously difficult. However, it is well known (and easy to prove) that for any integer x , there exist a least integer m and a least integer $n > m$ such that

$$g_m(x, s) \equiv g_n(x, s) \pmod{q} .$$

We denote this value of m by $\lambda = \lambda(x, s)$ and the value of $n - m$ by $\mu = \mu(x, s)$. λ is called the tail length, μ is called the cycle length and $\rho = \mu + \lambda = n$ is called the ρ -length of x with respect to $g(x, s)$ and q . Note that the values of $g_i(x, s)$ ($i = 0, 1, 2, \dots, \rho - 1$) are distinct modulo q , but that $g_\rho(x, s) \equiv g_\lambda(x, s) \pmod{q}$. Many probabilistic results are known concerning μ, λ, ρ for the iteration of random function (see Flajolet and Odlyzko [7] for several such results and references), and in [7] it is postulated that the properties of quadratic functions modulo an integer should be asymptotically the same as those of the class of all functions. Indeed, Bach [1] has proved that in initial stages, at least, quadratic functions do behave asymptotically like random functions. Thus, the expected values of μ, λ, ρ here should have values close to $\sqrt{\pi q/8}$, $\sqrt{\pi q/8}$ and $\sqrt{\pi q/2}$, respectively. Furthermore, the expected maximum values of μ, λ, ρ are respectively asymptotic to $c_1\sqrt{q}, c_2\sqrt{q}, c_3\sqrt{q}$, where $c_1 \approx .78248, c_2 \approx 1.73746, c_3 \approx 2.4149$. (See [7]).

Now consider the functional digraph of $g(x, s)$ over \mathbb{F}_q , the finite field of q elements. This is the directed graph whose nodes are the elements of \mathbb{F}_q and whose edges are the ordered pairs $\langle x, g(x, s) \rangle$ for all $x \in \mathbb{F}_q$. For example, the functional digraph of $g(x, 3)$ over \mathbb{F}_{11} is



Each connected component of the functional digraph contains exactly one cycle. Thus, in the case of $g(x, 9)$ over \mathbb{F}_{59} the functional digraph has exactly one component with its cycle containing the node 0.

We say that s is a *generator* for a prime q if for any integer x , there exists some minimal i (≥ 0) such that

$$g_i(x, s) = 0 \quad (2.5)$$

in \mathbb{F}_q . Thus, s is a generator for a prime q if and only if the functional digraph of $g(x, s)$ over \mathbb{F}_q has a single connected component whose cycle contains the node 0. If s is a generator, we define $\kappa (= \kappa(s)) = \max\{i\}$ of all the values of i given by (2.5). Thus, if s is a generator, then any $x \in \mathbb{F}_q$ can be written as

$$x = \pm \sqrt{s \pm \sqrt{s \pm \sqrt{s \pm \sqrt{\dots}}}}$$

with no more than κ radicals. Since the expected number of connected components for the functional digraph (under the same caveats as those mentioned above) is $(\log q)/2$ (see [7]), we would not expect to find many generators for a

given q and this is borne out by computations (see §4). Evidently, if $s = ab$ is a generator for some q , then $\omega(q) = q$ for $f(x) = ax^2 - b$ and $k > \kappa(s)$; the length of any corresponding Shanks chain can, therefore, not exceed $\kappa(s)$.

We now develop a technique for determining when s is a generator for a given q . Of course, this seems to be a very simple task because all we need do is start at some node $n_0 \in \mathbb{F}_q$ and by iterating g at this node compute the set \mathcal{S}_{n_0} of all distinct nodes in its tail and cycle \mathcal{C}_{n_0} over \mathbb{F}_q . If $\mathcal{S}_{n_0} = \mathbb{F}_q$, we are finished, but if $\mathcal{S}_{n_0} \neq \mathbb{F}_q$, we select $n_1 \notin \mathcal{S}_{n_0}$ and repeat the process. If for some h we get $\mathcal{C}_{n_0} = \mathcal{C}_{n_1} = \dots = \mathcal{C}_{n_h}$ and $\cup_{i=0}^h \mathcal{S}_{n_i} = \mathbb{F}_q$, then s must be a generator; otherwise s is not. The difficulty with this very simple algorithm is that when it is implemented on a computer and q is large, the amount of memory management required during its run greatly degrades its performance.

We can develop an algorithm for proving that s is a generator for a given q which does not involve a great deal of memory management if we are willing to do some extra work. To this end we define the following subsets of \mathbb{F}_q . We put $\mathcal{R}_0 = \{0\}$ and define \mathcal{R}_{i+1} recursively from \mathcal{R}_i by

$$\mathcal{R}_{i+1} = \{t : t^2 = r + s, r \in \mathcal{R}_i, t \neq 0\}.$$

For example, if $q = 367$ and $s = 1$, we have $\mathcal{R}_0 = \{0\}$, $\mathcal{R}_1 = \{\pm 1\}$, $\mathcal{R}_2 = \{\pm 288\}$, $\mathcal{R}_3 = \{\pm 17\}$, $\mathcal{R}_4 = \{\pm 237\}$, $\mathcal{R}_5 = \emptyset$.

We next establish some very simple results concerning \mathcal{R}_i .

Lemma 2.1. *If $g(x, s) \in \mathcal{R}_i$ and $x \neq 0$, then $x \in \mathcal{R}_{i+1}$.*

Proof. We have $x^2 = g(x, s) + s$. Since $x \neq 0$, we must have $x \in \mathcal{R}_{i+1}$ because $g(x, s) \in \mathcal{R}_i$.

Corollary. *If $g_j(x, s) = 0$ and $g_{j-i}(x, s) \neq 0$ ($0 \leq i \leq j$), then $x \in \mathcal{R}_i$.*

Proof. Follows easily by induction on j .

Lemma 2.2. *If $x \in \mathcal{R}_j$ ($j > 0$), then $g(x, s) \in \mathcal{R}_{j-1}$.*

Corollary. *If $x \in \mathcal{R}_j$ ($j > 0$), then $g_i(x, s) \in \mathcal{R}_{j-i}$ ($0 \leq i \leq j$).*

Theorem 2.3. *If $i > 0$, then*

$$\mathcal{R}_i = \{x : g_i(x, s) = 0, g_j(x, s) \neq 0 (j = 0, 1, 2, \dots, i-1)\}.$$

Proof. Follows easily from the corollaries of Lemmas 2.1 and 2.2.

Corollary 2.3.1. *If $j > i$, then $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$.*

Proof. If $x \in \mathcal{R}_j$, then $g_i(x, s) \neq 0$ ($i < j$), which means that $x \notin \mathcal{R}_i$.

Corollary 2.3.2. *If*

$$\sum_{i=0}^k \#\mathcal{R}_i = q , \quad (2.6)$$

then s is a generator for q . Conversely, if s is a generator for q , then (2.6) holds.

Corollary 2.3.2 can be used by a computer to prove that s is a generator for q , and, as we can produce \mathcal{R}_{i+1} from \mathcal{R}_i only and the values of $\#\mathcal{R}_i$ tend to be small, the memory requirements are modest. Of course, we must compute roughly $\#\mathcal{R}_i/2$ square roots modulo q to produce \mathcal{R}_{i+1} , but in practice, the Tonelli-Shanks algorithm (see Bach and Shallit [2], pp. 155-157) for doing this is very efficient; moreover, we developed a method based on the continued fraction expansion to compute square roots modulo q that is by roughly a factor of two faster than the Tonelli-Shanks algorithm (see §4). That the values of $\#\mathcal{R}_i$ tend to be relatively small follows on noting that in the case that (2.6) holds k must exceed the maximum λ value and is likely close to the maximum ρ value. Since we expect that about half of the values of r in \mathcal{R}_i are such that $((r+s)/q) = 1$, we expect that $\#\mathcal{R}_{i+1} \approx \#\mathcal{R}_i$. Thus, $k\#\mathcal{R}_i \approx q$ or $\#\mathcal{R}_i \approx q/k$ which will likely be less than \sqrt{q}/c_3 . However, it turns out that in practice the average value of $\#\mathcal{R}_i$ tends to be much smaller than \sqrt{q}/c_3 .

We can produce more useful necessary conditions for s to be a generator for q .

Theorem 2.4. *If s is a generator for q and $q > 2^j - 1$, then $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_j \neq \emptyset$.*

Proof. Since the degree of $g_i(x, s)$ as a polynomial in x is 2^i , there can be at most $\sum_{i=0}^{j-1} 2^i = 2^j - 1$ values of x in \mathbb{F}_q such that $g_i(x, s) = 0$ for $i \leq j-1$. Since $2^j - 1 < q$, there must (if s is a generator for q) be some x such that $g_k(x, s) = 0$ ($k > j-1$) and $g_i(s, x) \neq 0$ for all $0 \leq i \leq k-1$. Since $x \in \mathcal{R}_k$, we have $\mathcal{R}_k \neq \emptyset$ and therefore $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \dots, \mathcal{R}_j \neq \emptyset$.

From this result we readily conclude that 1 cannot be a generator for 367 because $367 > 2^8 - 1$ and $\mathcal{R}_5 = \emptyset$. Furthermore, if $(s/q) = -1$, then $\mathcal{R}_1 = \emptyset$ which means that s cannot be a generator for q . Note that this theorem provides a possible technique for eliminating a given s as a possible generator for q . If $q > 2^j - 1$ for some conveniently selected value of j , we need only compute the sets $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \dots, \mathcal{R}_j$. If we find an empty one, then we know that s cannot be a generator for q . If we consider the special case of $s = 1$, we know that $\mathcal{R}_1 = \{\pm 1\}$ and $\mathcal{R}_2 = \{x : x^2 \equiv 2 \pmod{q}\}$. We get $\mathcal{R}_2 = \emptyset$ if $(2/q) = -1$.

If $(2/q) = 1$, then $\mathcal{R}_3 = \{x : x^2 = 1+t, t^2 \equiv 2 \pmod{q}\}$. Now $(1-t)(1+t) = 1 - t^2 = -1$; hence, if $q \equiv 1 \pmod{8}$, then $((1-t)/q) = ((1+t)/q)$. Thus $\mathcal{R}_3 = \emptyset$ if $((1+t)/q) = -1$.

By a result of Barrucand and Cohn [3], we know this can only occur when q cannot be represented by the quadratic partition $a^2 + 32b^2$. We have the following theorem.

Theorem 2.5. *1 cannot be a generator for any prime q such that $q \equiv 3, 5 \pmod{8}$ or for any prime $q \equiv 1 \pmod{8}$ such that $q \neq a^2 + 32b^2$.*

3 Cycles

We mentioned earlier that it seems an unlikely event that any particular s will be a generator for a given q . Certainly, s will not be a generator if the functional digraph of $g(x, s)$ over \mathbb{F}_q contains a cycle which does not have 0 as a node. In this section we will investigate the problem of the existence of small cycles. We will consider the cases of minimal $\mu = 1, 2, 3, 4$ only. For $\mu = 1$, we want to know whether there exists some x such that $g(x, s) = x$ in \mathbb{F}_q . Clearly, this will be the case if and only if $((4s + 1)/q) = 0, 1$. Furthermore, this cycle is made up of the node x only when $(2x + 1)^2 \equiv 4s + 1 \pmod{q}$. Since $x \not\equiv 0 \pmod{q}$ under this condition, we have proved the following simple result.

Theorem 3.1. *If $((4s + 1)/q) = 0, 1$, then s is not a generator for q .*

We also have the following result.

Theorem 3.2. *If $s = t^2 - t$ ($\neq 0$), then s can never be a generator for any prime q .*

For the case that $\mu = 2$, we must have $g_2(x, s) = x$ and $g_1(x, s) \neq x$ in \mathbb{F}_q . Now it is easy to see that

$$g_2(x, s) - x = (g_1(x, s) - x)(x^2 + x - s + 1);$$

thus, we have a cycle of length 2 if and only if $((4s - 3)/q) = 0, 1$. However, since this cycle is made up of the nodes $\{x, -1 - x\}$ when $(2x + 1)^2 \equiv 4s - 3 \pmod{q}$, we see that 0 is not in the cycle only when $s \not\equiv 1 \pmod{q}$.

Theorem 3.3. *If $s \not\equiv 1 \pmod{q}$ and $((4s - 3)/q) = 0, 1$, then s is not a generator for q .*

Also, if $s = t^2 + t + 1$, then s is not a generator for any q unless $s \equiv 1 \pmod{q}$ and 1 is a generator for q .

Theorem 3.4. *If $s = t^2 + t + 1$, ($t \leq B$), then s can never be a generator for any prime q if $t \not\equiv 0, -1 \pmod{r}$ for all primes $r \leq B$ such that 1 is a generator for r .*

Now 1 is a generator for $r = 3, 7, 23$ and for no other prime ≤ 5000 . Thus, if $s = t^2 + t + 1$ ($t \leq 5000$) and $t \not\equiv 0, -1 \pmod{3}$, $t \not\equiv 0, -1 \pmod{7}$ and $t \not\equiv 0, -1 \pmod{23}$, then s can never be a generator for any prime q .

Notice that for the forms of s given by Theorems 3.2 and 3.4 we can never have $\omega(q) = q$ when $s = ab$. Thus, we should (under Conjecture H) be able to find corresponding Shanks chains of arbitrary length. Also for these forms we have solutions in \mathbb{Z} of $g_i(x, s) = x$. However, by a result of Narkiewicz ([14], Theorem 2), we know that there are no values of x in \mathbb{Z} such that the least cycle length of $g(x, s)$ exceeds 2. This suggests (but certainly does not prove) that the only possible values of s which can never be a generator for any q are those given in Theorems 3.2 and 3.4.

We next consider the case of cycles of length 3 over \mathbb{F}_q . This has been examined for arbitrary fields of odd characteristic by Morton [12], but we will use a somewhat different and computationally more convenient approach here. We have

$$g_3(x, s) - x = (g_1(x, s) - x)c(x) ,$$

where $c(x) =$

$$x^6 + x^5 + (1-3s)x^4 + (1-2s)x^3 + (1-3s+2s^2)x^2 + (1-2s+s^2)x + 1 - s + 2s^2 - s^3 .$$

If we have a cycle of length 3 for $g(x, s)$ over \mathbb{F}_q , we must have three distinct zeros of $c(x)$, because if x is a zero of $c(x)$ in \mathbb{F}_q , then so also must be $g_1(x, s)$ and $g_2(x, s)$. That is, $x^2 - s$ and $(x - s)^2 - s$ must be zeros if x is. This follows from the simple observation that $g_{i+j}(x, s) = g_i(g_j(x, s), s)$. By an old result of Escott [6], we know that if $c(x)$ is to have a zero in \mathbb{F}_q , then $s = a^2 - a + 2$ for some $a \in \mathbb{F}_q$. When this happens, we get

$$c(x) = (x^3 + ax^2 + (a-s-1)x - (as-s+1))(x^3 + a'x^2 + (a'-s-1)x - (a's-s+1)) ,$$

where $a' = 1 - a$. Then, a necessary condition that there be a cycle of length 3 for $g(x, s)$ over \mathbb{F}_q is that $((4s-7)/q) = 0, 1$. Now if $q \neq 3$ and $s = a^2 - a + 2$, we see that

$$p(x) = x^3 + ax^2 + (a-s-1)x - (as-s+1)$$

will have 3 zeros in \mathbb{F}_q if and only if

$$h(x) = x^3 + 3(2a-1-4s)x - (16as+2a-20s+13)$$

has 3 zeros in \mathbb{F}_q . Since

$$16as+2a-20s+13 = (4a-3)(4s-2a+1) ,$$

it is a simple matter to evaluate the discriminant D of $h(x)$ as

$$D = -27(4s-2a+1)^2 .$$

If $4s-2a+1 = 0$ in \mathbb{F}_q , then $p(x)$ has the zero $-a/3$ with multiplicity 3; furthermore, if $x = -a/3$, then $g(x, s) = x$. Thus, we exclude this possibility and we find that $(D/q) = (-3/q) \equiv q \pmod{3}$. By classical results concerning the solubility of cubic congruences modulo q (see Dickson [5], p. 256) we know that $h(x)$ can have 3 zeros in \mathbb{F}_q if and only if $q \equiv (D/q) \pmod{3}$ and

$$\alpha^{(q^2-1)/3} \equiv 1 \pmod{q} ,$$

where $\alpha = (A + \sqrt{D})/2$, $A = 16as+2a-20s+13$. We note that $\alpha = (4s-2a+1)\gamma$, where γ is a zero of $x^2 - (4a-3)x + 4s-2a+1$. If ζ is a primitive cube root of unity ($\zeta^2 + \zeta + 1 = 0$), then $\gamma = 2a + 3\zeta$.

If $q \equiv -1 \pmod{3}$, then $\alpha^{(q^2-1)/3} \equiv [\lambda/q] \pmod{q}$, where $[\lambda/q]$ is defined to be the value of ζ^i in $\mathbb{Z}[\zeta]$ such that $\gamma^{(q^2-1)/3} \equiv \zeta^i \pmod{q}$. If $q \equiv 1$

$(\text{mod } 3)$, then $\alpha^{(q^2-1)/3} = [\alpha/\pi]$, where π is a primary prime factor of q in $\mathbb{Z}[\zeta]$ and $[\alpha/\pi]$ is that value of ζ^i such that $\alpha^{(q^2-1)/3} \equiv \zeta^i \pmod{\pi}$. Thus, if q does not divide $4s - 2a + 1$ and $q \neq 3$, then $p(x)$ has three zeros in \mathbb{F}_q if and only if $[\gamma/q] = 1$ when $q \equiv -1 \pmod{3}$ or $[(4s - 2a + 1)\gamma/\pi] = 1$ when $q \equiv 1 \pmod{3}$ and π is a primary prime factor of q . Also, if a cycle of length 3 exists for this s and a and $as - s + 1 \neq 0$ in \mathbb{F}_q , then none of the zeros of $p(x)$ can be zero. We have proved the following theorem.

Theorem 3.5. *If q is a prime, $q \neq 3$, q does not divide $4s - 2a + 1$ and $((4s - 7)/q) = 0, 1$, then there is a cycle of length 3 for $g(x, s)$ over \mathbb{F}_q if and only if $s \equiv a^2 - a + 2 \pmod{q}$, $\gamma = 2a + 3\zeta$ and $[\gamma/q] = 1$ when $q \equiv -1 \pmod{3}$ or $[(4s - 2a + 1)\gamma/\pi] = 1$ when $q \equiv 1 \pmod{3}$ and π is a primary prime divisor of q . Furthermore, if a cycle of length 3 exists for this value of s and a , and q does not divide $as - s + 1$, then s cannot be a generator for q .*

We remark here that the values of $[\xi/\psi]$ here can be obtained quite rapidly by making use of the idea of Jacobi (see Williams and Holte [21]).

If we consider the case of $s = 1$, we see that we must have $(-3/q) = 1$ in order to have $\mu = 3$. Thus, we require that $q \equiv 1 \pmod{3}$ and we may put

$$4q = L^2 + 27M^2 \quad (L \equiv 1 \pmod{3})$$

as an essentially unique (up to the sign of M) quadratic partition of $4q$. Since $a^2 - a + 1 \equiv 0 \pmod{q}$ is soluble for $a \in \mathbb{Z}$, we may put $q = \pi_1\pi_2$, where $\pi_1 | a + \zeta$, $\pi_2 | a - 1 - \zeta$ and π_1, π_2 are primary prime divisors of q in $\mathbb{Z}[\zeta]$. We have

$$(4s - 2a + 1)\gamma \equiv (5 - 2a)(2a + 3\zeta) \equiv \zeta(5 + 2\zeta) \equiv 3\zeta - 2 \pmod{\pi_1}.$$

Also, $4s - 2a + 1 = 5 - 2a \neq 0$ if $q \neq 19$. Since $2 - 3\zeta$ is a primary prime divisor of 19, we get

$$\left[\frac{\gamma(4s - 2a + 1)}{\pi_1} \right] = \left[\frac{3\zeta - 2}{\pi_1} \right] = \left[\frac{2 - 3\zeta}{\pi_1} \right] = \left[\frac{\pi_1}{2 - 3\zeta} \right]$$

by the law of cubic reciprocity. Now we select the sign of M such that $\pi_1 = (L + 3M)/2 + 3M\zeta$ and note that $\zeta \equiv -12 \pmod{2 - 3\zeta}$; hence, $\pi_1 \equiv (L + 3M)/2 - 36M \pmod{2 - 3\zeta}$. Thus, $[\pi_1/(2 - 3\rho)] = 1$ if and only if

$$((L + 3M)/2 - 36M)^6 \equiv 1 \pmod{19}.$$

Since $2^{18} = 8^6 \equiv 1 \pmod{19}$, this is equivalent to $(4L + 9M)^6 \equiv 1 \pmod{19}$ or $4L + 9M \in \{\pm 1, \pm 7, \pm 8\} \pmod{19}$. We now have the following result for $s = 1$.

Theorem 3.6. *If $s = 1$ and $q \neq 3, 19$, there is a cycle of length 3 for $g(x, s)$ over \mathbb{F}_q if and only if $q \equiv 1 \pmod{3}$, $4q = L^2 + 27M^2$ and either $4L + 9M$ or $4L - 9M \in \{\pm 1, \pm 7, \pm 8\} \pmod{19}$. If both $4L + 9M$ and $4L - 9M \in \{\pm 1, \pm 7, \pm 8\} \pmod{19}$, then there are exactly two cycles of length 3. Also, 1 cannot be a generator for any q such that $g(x, s)$ over \mathbb{F}_q has a cycle of length 3.*

If we consider the example of $q = 157$, we get $L = -14$, $M = \pm 4$. Since $4 \cdot (-14) + 9 \cdot 4 \equiv -1$ and $4 \cdot (-14) - 9 \cdot 4 \equiv 3 \pmod{19}$, we see that there is a single cycle for $s = 1$; this is $\{92, 142, 67\}$. If $q = 151$, we get $L = 19$, $M = \pm 3$ and $4L \pm 9M \equiv \pm 8 \pmod{19}$. Thus, in this case we get two cycles, namely $\{19, 58, 41\}$ and $\{85, 127, 122\}$.

For the case of $\mu = 4$, it is convenient to use the techniques of Morton ([13], pp. 91-92). While they were employed with respect to \mathbb{Q} , they are very readily applicable to the case of \mathbb{F}_q . With some very simple manipulation of his formulas, it is easy to derive the following theorem.

Theorem 3.7. *If q is an odd prime and q does not divide $(4s-5)(16s^2+8s+5)$, there can be a cycle of length 4 for $g(x, s)$ over \mathbb{F}_q if and only if there exists some solution $z \in \mathbb{Z}$ of*

$$z^3 + (3 - 4s)z + 4 \equiv 0 \pmod{q}$$

and a corresponding solution $w \in \mathbb{Z}$ of

$$w^2 - zw - 1 \equiv 0 \pmod{q}$$

such that

$$(z(zw + 2)(z + 2)/q) = 1.$$

Furthermore, if such a cycle exists and q does not divide $z^6 + 2z^5 + 4z^4 + 6z^3 - 5z^2 - 8z - 16$, then s cannot be a generator for q .

As an example, we give $q = 23$ and $s = 40$. We find $z = 38$ and $w = 2$ and the corresponding cycle is $\{36, 15, 39, 21\}$.

4 Algorithms and Computational Results

We have already mentioned that one would not expect to find many generators s for a given prime q . In fact, by using a computer, we found all the generators for each prime $< 10^4$. Let $n(q)$ denote the number of generators for q and

$$N(x) = \#\{q : n(q) = x, q < 10^4\}.$$

In Table 2 we present some values of $N(x)$; note that $N(x) = 0$ if $n > 8$.

Table 2.

x	$N(x)$	x	$N(x)$	x	$N(x)$
0	378	3	100	6	0
1	464	4	25	7	0
2	258	5	3	8	1

Thus, the average number of generators for each prime $q < 10^4$ is $\sum_0^8 iN(i) / \sum_0^8 N(i) \approx 1.14$. Incidentally, the value of q for which there are 8 generators is $q = 9767$ with generators 1051, 1937, 2217, 2301, 3478, 3697, 5471, 6803.

We used these generators to sieve out all the values of $s < 1000$ which must be a generator for some $q < 10^4$. The remaining values of s are presented in the tableau below.

2	3	6	12	20	21	30	42	56	72	90
105	108	110	111	128	132	156	182	195	198	
206	210	213	215	240	251	272	273	287	290	293
303	306	311	338	342	356	380	381			
420	437	462	471	483	495					
506	525	545	548	552	570	591	593			
600	612	623	630	642	650	651	656	657	675	
702	713	723	726	735	740	752	755	756	768	770
800	812	821	840	857	861	870				
908	912	930	936	957	965	987	992	993	996	

Of these remaining numbers, we see that 2, 3, 6, 12, 20, 21, 30, 42, 56, 72, 90, 110, 111, 132, 156, 182, 210, 240, 272, 273, 306, 342, 380, 381, 420, 462, 506, 552, 600, 650, 651, 702, 756, 812, 870, 930, 992 and 993 are all of the forms given in Theorems 3.2 and 3.4; hence, these 38 numbers can never be generators for any q . This leaves 54 numbers (< 1000) which may be generators for some prime $q > 10^4$. If, in fact, the only values of s which can never be generators for any q are those given by Theorems 3.2 and 3.4, we would expect to be able to eliminate all of these 54 remaining numbers by increasing our limit on q beyond 10^4 . This means that we need to develop reasonably efficient algorithms for detecting when a particular s is a generator for a given q .

Such an algorithm is implicit in Corollary 2.3.2, but as we have seen earlier, it is most unlikely that a given s will be a generator for a given q ; thus, it is best to develop an algorithm that will quickly determine that s is not a generator for q (when this is the case).

We note that since the average cycle length is expected to be $\sqrt{\pi q/8}$, the chance that it contains a zero node is $\sqrt{8/\pi q}$ which is very small. Indeed, if we examine all primes q such that $109 \leq q < 10^5$ for $s = 108$ with the property that $(s/q) = 1$ and $(4s + 1/q) = -1$ and $(4s - 3/q) = -1$ and let $m(x) = \#\{q : x$ is the smallest number ≥ 0 such that the cycle in the component beginning with x does not contain a zero node}, when $m(x) \neq 0$ we get Table 3.

Thus, to determine that s is not a generator for q we have the following algorithm: For $x = 0, 1, 2, \dots$ up to a certain bound B we check whether the cycle in the component beginning with x contains a zero node. For this, we compute the sequences $(g_i(0, s))_{i \geq 0}$ and $(g_{2i}(0, s))_{i \geq 0}$, and for each $i = 1, 2, \dots$ we check whether $g_i(0, s) \equiv g_{2i}(0, s) \pmod{q}$. This is expected to happen for $i \approx 1.0308\sqrt{q}$ (Floyd's method, see [15]). When this is the case, we know that $g_i(0, s)$ is in the cycle for that i , so that we compute $g_{i+1}(0, s), g_{i+2}(0, s), \dots$ until we find a minimal j such that $g_{i+j}(0, s) \equiv 0 \pmod{q}$ or $g_{i+j}(0, s) \equiv g_i(0, s)$.

Table 3.

x	$m(x)$	x	$m(x)$
0	1185	5	1
1	11	7	1
2	4	9	1
4	2	45	1

(mod q). If the latter happens first, we know that the zero node is not in the cycle, and s is not a generator for q . Otherwise, we compute for $x = 1, 2, \dots$ the sequences $(g_i(x, s))_{i \geq 0}$ and $(g_{2i}(x, s))_{i \geq 0}$ and, while doing this, check whether $g_{2i-1} \equiv 0 \pmod{q}$ or $g_{2i}(x, s) \equiv 0 \pmod{q}$. As soon as this happens, we know that x belongs to the same component as the zero node, which is in the cycle. If $g_i(x, s) \equiv g_{2i}(x, s) \pmod{q}$ for some i and $g_j(x, s) \not\equiv 0 \pmod{q}$ for all $j \leq 2i$, we know that x belongs to a component different from the one with the zero node. As soon as this happens for some x , we know that s is not a generator for q . Otherwise, s may be a generator.

For example, for the 54 remaining values for s and the primes q between 10^4 and 10^5 for which $(s/q) = 1$ and $(4s + 1/q) = -1$ and $(4s - 3/q) = -1$, with $B = 100$ we find that altogether 40 pairs (s, q) pass this test. Among these 40 pairs, there are 15 cases where s indeed is a generator for q . When working with $B = 1000$, only 17 pairs (s, q) pass this test. Notice that, for given s , the test has to be applied only to about 12.5% of the 8363 primes between 10^4 and 10^5 , since about 87.5% of the q are eliminated by checking the three Legendre symbols.

The running time for each $x = 0, 1, \dots, B$ is proportional to the ρ -length and hence is expected to grow with \sqrt{q} . Given s , for larger values of q we therefore use the following, faster algorithm: First we choose parameters j_0 and B . Now, given a pair (s, q) such that $(s/q) = 1$ and $(4s + 1/q) = -1$ and $(4s - 3/q) = -1$, we check whether \mathcal{R}_3 is empty. If this is the case, s cannot be a generator for q . Otherwise, we apply the criteria provided by Theorems 3.5 and 3.7. If after that it is still possible that s is a generator for q , we compute $\mathcal{R}_4, \dots, \mathcal{R}_{j_0}$. Only if $\mathcal{R}_j \neq \emptyset$ for all $k \leq j_0$ and if $\sum_{j=0}^{j_0} \#\mathcal{R}_j < q$, we check for $x = 0, 1, 2, \dots, B$ whether the cycle in the component beginning with x contains a zero node. For $q \in [10^6, 10^{10}]$, a suitable choice for j_0 and B is $j_0 = 30$ and $B = 2000$. To illustrate the performance of this algorithm, for $s = 108, 840$ and $n = 4, 5, 6, 7, 8, 9$ we consider the least 10000 primes $\geq 10^n$ and let $k(n, s) = \#\{q : (s/q) \neq -1 \text{ or } (4s + 1/q) \neq -1 \text{ or } (4s - 3/q) \neq -1\}$. By $r_3(n, s)$ we denote the number of those primes q not included in $k(n, s)$ and for which $\mathcal{R}_3 = \emptyset$. By $c_3(n, s)$ we denote the number of those q among the primes not counted so far for which we can prove the existence of a cycle of length 3 using Theorem 3.5. Then, by $c_4(n, s)$ we denote the number of primes among the remaining values of q for which Theorem 3.7 establishes the existence of a cycle of length 4. By $r_{30}(n, s)$ we denote the number of those q that passed all tests so far and for which one of the $\mathcal{R}_4, \dots, \mathcal{R}_{30}$ is empty and $\sum_{j=0}^{30} \#\mathcal{R}_j < q$.

Then by $b_{2000}(n, s)$ we denote the number of those remaining q which do not pass the very last test that checks whether the component beginning with x ($x = 0, 1, \dots, 2000$) contains a zero node. Our sample results are shown in Table 4. Here, the last column indicates how many primes q have passed all tests. These values of q are the only remaining candidates for which s can be a generator.

Table 4.

n	$k(n, s)$	$r_3(n, s)$	$c_3(n, s)$	$c_4(n, s)$	$r_{30}(n, s)$	$b_{2000}(n, s)$	survivors
$s = 108$							
4	5019 + 2506 + 1224	516	208	103	342	82	0
5	4993 + 2456 + 1299	507	238	103	327	77	0
6	4983 + 2512 + 1230	524	192	119	345	95	0
7	5019 + 2488 + 1265	496	192	99	350	91	0
8	5015 + 2518 + 1216	489	192	122	366	82	0
9	5039 + 2497 + 1208	510	204	115	360	67	0
$s = 840$							
4	4985 + 2534 + 1257	512	207	108	320	77	0
5	5042 + 2532 + 1211	452	214	112	360	77	1 ($q = 182101$)
6	4971 + 2517 + 1234	487	215	115	376	85	1 ($q = 1053583$)
7	5040 + 2525 + 1245	442	186	145	346	71	0
8	4990 + 2508 + 1245	488	214	112	354	89	0
9	4942 + 2558 + 1288	478	217	98	346	73	0

To process the values of q which remain in the end, we choose some larger bound B' and check for $x = 2001, \dots, B'$ whether all components that begin with x end in a cycle that contains the zero node. For $q = 182101$ in Table 4 we find, for example, that the least positive number x that leads to a cycle that does not contain the zero node is $x = 3972$, while for $q = 1053583$ this is the case for $x = 80173$. Hence, $s = 840$ is not a generator for these values of q .

With the algorithm prescribed above we examined all primes q such that $100003 \leq q \leq 2 \cdot 10^8$ for all 54 remaining values of s . For 36 values of s we found a prime q such that s is a generator for q . In Table 5 we give the corresponding values of s and q . Notice that for each s , we only consider the least possible prime q for which s is a generator. In the third and sixth columns we also indicate the minimal value for k such that $\sum_{i=0}^k \#\mathcal{R}_i = q$, which also is the least k such that \mathcal{R}_{k+1} is empty.

In summary of Table 5, Table 6 shows for various ranges of q the number of values of s such that q is the least prime for which s is a generator.

The remaining 18 values of s which are not a generator for any prime $q < 2 \cdot 10^8$ are 108, 128, 290, 338, 495, 525, 545, 623, 630, 656, 675, 723, 735, 755, 770, 800, 936, 987. Extrapolating from the data in Table 6, we expect that in order to find appropriate values of q for all of these values of s , we would have to examine all values of q up to at least 10^{11} . This seems to be a hopeless task at present because we know of no algorithm that executes in fewer than constant times q

Table 5.

s	q	k	s	q	k
105	97729	863	593	92802097	21661
195	1956979	2658	612	134639	833
198	820361	1682	642	1643779	3607
206	1746581	4191	657	23035711	11086
213	5994631	6548	713	275657	1242
215	25847	508	726	14087	286
251	89231	582	740	16691	374
293	56891	835	752	23059	444
287	8207041	6542	768	12441217	6044
303	29947	363	821	40682441	9916
311	631723	2063	840	10830383	6669
356	493853	1470	857	98947	888
437	34283	359	861	29947427	15438
471	20347	369	908	2060843	2695
483	31228199	17327	912	141184027	26912
548	58991	708	957	1686701	3379
570	493811	1757	965	39191	629
591	11369	204	996	35053	367

Table 6.

range for q	# generators
$[10^4, 10^5]$	15
$[10^5, 10^6]$	6
$[10^6, 10^7]$	7
$[10^7, 10^8]$	7
$[10^8, 2 \cdot 10^8]$	1

steps for proving that s is a generator for a certain prime q . Notice that, taking into account the 38 numbers of the forms given in Theorems 3.2 and 3.4 which cannot be a generator for any q , we therefore are left with 56 values of $s \leq 1000$ for which a corresponding Shanks chain might have arbitrary length.

The largest value of q that appears in Table 5 is $q = 141184027$, which is the least prime for which $s = 912$ is a generator. Here, $\sum_{i=0}^k \#\mathcal{R}_i = q$ for $k = 26912$, with \mathcal{R}_{26913} being the first set that is empty. The maximum value for $\#\mathcal{R}_i$ is 13198, while the average value for $\#\mathcal{R}_i$ is 5245. With an implementation using the computer algebra system LiDIA [11], the computation of the \mathcal{R}_i took 4 hours, 6 minutes and 54 seconds on a SPARC Ultra-60. For this computation we used the Tonelli-Shanks algorithm to compute the square root of a modulo q .

We also developed another method to compute the square root of a modulo q that makes use of the continued fraction expansion of a/q . It works as follows. If $q \equiv 1 \pmod{4}$, i.e., $(-1/q) = 1$, let λ such that $\lambda^2 \equiv -1 \pmod{q}$ and let n be a quadratic non-residue of q (which we can easily find by trial). If $q \equiv -1$

(mod 4), let $\lambda = 1$ and $n = -1$. Now for $1 \leq x < \sqrt{q}$ we precompute tables of the Legendre symbols (x/q) and of the values $G(x)$ and $H(x)$, where

$$G(x) \equiv \begin{cases} \sqrt{x} & \text{if } (x/q) = 1 \\ \sqrt{nx} & \text{if } (x/q) = -1 \end{cases} \pmod{q}$$

and

$$H(x) \equiv \begin{cases} 1/\sqrt{x} & \text{if } (x/q) = 1 \\ 1/\sqrt{nx} & \text{if } (x/q) = -1 \end{cases} \pmod{q};$$

Here, \sqrt{t} denotes either of the solutions, when they exist, of $y^2 \equiv t \pmod{q}$. To compute the square root of a modulo q when $(a/q) = 1$, we put $r_0 = q$, $r_1 = a$, $B_1 = 0$, and $B_2 = 1$. For $i \geq 2$ we let

$$\begin{aligned} r_i &= r_{i-2} \pmod{r_{i-1}}, \\ q_{i-1} &= r_{i-2} \pmod{r_{i-1}}, \\ B_{i+1} &= q_{i-1}B_i + B_{i-1}, \end{aligned}$$

until we find a minimal i such that $B_{i+1} > \sqrt{q}$. Then

$$(-1)^i r_{i-1} \equiv aB_i \pmod{q}$$

and

$$(a/q) = (B_i/q)(r_{i-1}/q)(-1/q)^i,$$

where $B_i, r_{i-1} < \sqrt{q}$ (see [20]). It is now easy to verify that if $(a/q) = 1$, then

$$a \equiv \begin{cases} (G(r_{i-1})H(B_i))^2 & \pmod{q} \quad \text{if } 2 \mid i \\ (\lambda G(r_{i-1})H(B_i))^2 & \pmod{q} \quad \text{otherwise.} \end{cases}$$

Thus, once B_i and r_{i-1} have been found, (a/q) can be easily determined and, if $(a/q) = 1$, then $\sqrt{a} \pmod{q}$, can be computed simply by table look-ups and multiplication modulo q . It turns out that for large values of q , this method speeds up our algorithm by about a factor of 2. For example, to prove that $s = 912$ is a generator for $q = 141184027$ with the new method took only 2 hours, 6 minutes and 3 seconds, on the same machine as before.

We made a special effort to find values of q for which $s = 108$ and $s = 290$ are generators. This was without success – we only found that $s = 108$ and $s = 290$ are not generators for any prime $q < 10^{10}$. Moreover, because of Theorem 3.4 we also tried to find other values of q for which $s = 1$ is a generator. We found that 1 is a generator for 3, 7, 23, 19207 and no other prime $< 2.1 \cdot 10^9$.

While for most of the primes q we can determine very rapidly that a given s is not a generator for q , once in a while we run into a value for q which requires much more effort. For example, for $s = 1$ and $q = 1523053897$, the least positive x such that the component beginning with x ends in a cycle that does not contain the zero node is $x = 2765848$; this component consists of 962 elements, and the corresponding cycle length is 26. On a SPARC Ultra-60, it took 14 days, 16 hours and 35 minutes to find this value of x . To determine that s is not a generator

for q by considering the sets \mathcal{R}_i , we have to compute \mathcal{R}_i up to $i = 121092$ (i.e., \mathcal{R}_{121093} is the first set that is empty) until we find that the component that contains the zero node consists of only $1523052733 = q - 1164$ elements. This computation took 1 day, 9 hours and 56 minutes on a SPARC Ultra-60.

In Table 7, we list for $s = 1$, $s = 108$ and $s = 290$ those primes $q > 10^9$ ($q < 2 \cdot 10^9$ for $s = 1$, and $q < 10^{10}$ for $s = 108, 290$) that have passed all tests described prior to Table 4, and for which all $x = 0, 1, \dots, 10000$ end in a cycle that contains the zero node. The respective cycle lengths are indicated in the second column of Table 7. However, for all these primes we eventually found a second component: in the third column we give the least x that ends in a cycle that does not contain the zero node, while the last column shows the corresponding cycle length.

Table 7.

q	μ	x	μ_x
$s = 1$			
1055114873	2	53870	70
1121788583	2	12934	278
1307586407	2	11064	41
1523053897	2	2765848	26
$s = 108$			
1361042663	35227	99379	7
3323409469	130529	130529	905
3570912959	28182	1574734	24
3945934931	79534	694699	13
5626917623	159574	47016	373
$s = 290$			
1492251769	78613	69682	123
2258948569	97638	18895	1198
2262261047	26405	62921	18
3870012343	153977	190103	24657
4696002397	19510	29918	350
5824284551	42637	37610	499
7865621479	98782	42567	379
8273290073	174261	1627204	44

Another exceptional pair (s, q) is given by $s = 545$ and $q = 16251619$: Here, we find that the component that ends in a cycle with the zero node consists of $q - 12$ elements, while there is a second component that consists of 12 elements and ends in a cycle of length 5. The least positive number x that belongs to that second component is $x = 4048245$.

References

1. E. Bach, *Toward a theory of Pollard's rho method*, Information and Computation **90** (1991), 139–155.
2. E. Bach and J. O. Shallit, *Algorithmic number theory*, vol. Vol. 1, MIT Press, 1996.
3. P. Barrucand and H. Cohn, *A note on primes of type $x^2 + 32y^2$, class number and residuacity*, J. Reine Angew. Math. **238** (1969), 67–70.
4. P. T. Bateman and R. Horn, *Primes represented by irreducible polynomials in one variable*, Proc. Sympos. Pure Math., Vol VIII (Providence), AMS, 1965, pp. 119–132.
5. E. Dickson, *History of the theory of numbers*, vol. 1, Chelsea, New York, 1952.
6. E. B. Escott, *Cubic congruences with three real roots*, Annals of Math. **11** (1909–10), no. 2, 86–92.
7. P. Flajolet and A. M. Odlyzko, *Random mapping statistics*, Advances in Cryptology - EUROCRYPT '89, LNCS, vol. 434, 1990, pp. 329–354.
8. T. Forbes, *Prime clusters and Cunningham chains*, Math. Comp. **68** (1999), no. 228, 1739–1747.
9. R. K. Guy, *Unsolved problems in number theory*, second edition ed., Springer-Verlag, Berlin, 1994.
10. G. H. Hardy and J. E. Littlewood, *Some problems of partitio numerorum. III. On the expression of a number as a sum of primes*, Acta Math. **44** (1923), 1–70.
11. LiDIA Group, Technische Universität Darmstadt, Darmstadt, Germany, *LiDIA - a library for computational number theory, version 1.3*, 1997.
12. P. Morton, *Arithmetic properties of periodic points of quadratic maps*, Acta Arithmetica **62** (1992), 343–372.
13. P. Morton, *Arithmetic properties of periodic points of quadratic maps II*, Acta Arithmetica **87** (1998), 89–102.
14. W. Narkiewicz, *Polynomial cycles in algebraic number fields*, Colloq. Math. **58** (1989), 151–155.
15. J. M. Pollard, *A Monte Carlo method for factorization*, BIT **15** (1975), no. 3, 331–335.
16. A. Schinzel and W. Sierpiński, *Sur certaines hypothèses concernant les nombres premiers*, Acta Arith. **4** (1958), 185–208.
17. A. Schinzel and W. Sierpiński, *Remarks on the paper "Sur certaines hypothèses concernant les nombres premiers"*, Acta Arith. **7** (1961), 1–8.
18. D. Shanks, Letter to D.H. and Emma Lehmer, June 10, 1969.
19. D. Shanks, *Supplemental data and remarks concerning a Hardy-Littlewood conjecture*, Math. Comp. **17** (1963), 188–193.
20. A. Stein and H. Williams, *An improved method of computing the regulator of a real quadratic function field*, Algorithmic Number Theory Seminar ANTS-III, Lecture Notes in Computer Science, vol. 1423, Springer-Verlag, 1998, pp. 607–620.
21. H. C. Williams and R. Holte, *Computation of the solution of $x^3 + dy^3 = 1$* , Math. Comp. **31** (1977), 778–785.

Asymptotically Fast Discrete Logarithms in Quadratic Number Fields

Ulrich Vollmer*

Technische Universität Darmstadt, Institut für Theoretische Informatik
Alexanderstr. 10, 64283 Darmstadt
uvollmer@cdc.informatik.tu-darmstadt.de

Abstract. This article presents algorithms for computing discrete logarithms in class groups of quadratic number fields. In the case of imaginary quadratic fields, the algorithm is based on methods applied by Hafner and McCurley [HM89] to determine the structure of the class group of imaginary quadratic fields. In the case of real quadratic fields, the algorithm of Buchmann [Buc89] for computation of class group and regulator forms the basis. We employ the rigorous elliptic curve factorization algorithm of Pomerance [Pom87], and an algorithm for solving systems of linear Diophantine equations proposed and analysed by Mulders and Storjohann [MS99]. Under the assumption of the Generalized Riemann Hypothesis, we obtain for fields with discriminant d a rigorously proven time bound of $L_{|d|}[\frac{1}{2}, \frac{3}{4}\sqrt{2}]$.

1 Introduction

Currently the best available algorithms for extracting discrete logarithms (DL) in class groups of number fields proceed by computing the class group—generators and relations—first, and continue from there by linear algebra to calculate the DL (cf. e.g. [BD90]). This seems more work than necessary, since the class group problem appears to be more difficult than the DL problem. Indeed, we will show in this article that it is possible to evade the necessity of computing the class group first. On the contrary, the methods applied can be extended to compute the class group.

Since Gauss, there has been continuous interest in computing class groups. Comparatively recently, the interest has also turned to the question of how to compute them *efficiently*. With [HM89], and [Buc89] we would like to mention two papers which made a breakthrough by proving that this calculation can be done in time subexponential in the size of the discriminant of the field: [HM89] did this for imaginary quadratic fields, [Buc89] for general number fields. Here, we will employ the methods of these papers, and sharpen their results.

The point of view of this article, however, is cryptographic, and we will focus on giving rigorous time bounds for the DL problem. The cryptographic interest in this problem arises from several proposals for using class groups as the underlying algebraic structure for cryptosystems, see e.g. [BW88], and [McC89].

* research supported by the DFG

In finite fields—the algebraic structure for which DL-based cryptosystems were first proposed—discrete logarithms can be computed in time $L_q[\frac{1}{2}, 1]$, where q is the size of the field, and, for real numbers $x > e$, a , b , we set as usual $L_x[a, b] = \exp(b(\log x)^a(\log \log x)^{(1-a)})$.

When we try to reproduce this result and the methods which were used to obtain it in the case of class groups we have to struggle with the fact that the size of the group we are working with is not known beforehand. It is therefore tempting to overcome this difficulty by first computing this size or even the structure of the group at hand, and then use standard procedures for general groups to solve the DL problem.

Our approach evades this necessity. We do apply the usual index calculus approach, as proposed in the case of imaginary quadratic fields by Hafner and McCurley [HM89] together with an improvement already proposed by Hafner and McCurley themselves, namely the use of the rigorous elliptic curve factorization algorithms introduced by Pomerance [Pom87]. However, the final linear algebra step, which originally involved the calculation of the Hermite normal form of the relation matrix is replaced by a direct computation of the discrete logarithm on the basis of the relation matrix, employing the recently proposed algorithm for solving Diophantine linear systems by Mulders and Storjohann [MS99].

As a result of these improvements we prove rigorously that our algorithms (DL extraction in the imaginary and real quadratic case, class group computation in the imaginary quadratic case) can be executed in time $L_{|d|}[\frac{1}{2}, \frac{3}{4}\sqrt{2}+o(1)]$, with the sole premise of the Generalized Riemann Hypothesis (GRH). In comparison to McCurley [McC89] who obtained the same time bound for the computation of the class group of an imaginary quadratic field we do not have to make any assumptions on the behavior of intermediate results.

For background information on quadratic fields, and their class groups we refer the reader to [LP92] and [Coh93].

2 The Imaginary Quadratic Case – Description of the Algorithm

Let K be an imaginary quadratic field of discriminant $-d$. For ease of calculation we will work with binary forms instead of ideals. We denote by $Cl(-d)$ the set of $\text{PSL}_2(\mathbb{Z})$ -equivalence classes of positive definite primitive binary quadratic forms of discriminant $-d$, with group structure induced by Gaussian composition. The class with representative f will be denoted by $[f]$. We will omit the brackets, however, where no confusion will arise.

The IQNF-DL problem. Given two forms g and h the task is to decide whether there exists an exponent $l \in \mathbb{Z}$ such that $[g]^l = [h]$, and, if the answer is positive, to find one. (Since this task is trivial if g is in the principal class—which can easily be checked by reduction—we will assume in the following that it is not.)

For the set-up of the index calculus we first need a large set \mathcal{F} of prime forms as a factor base. Let $\mathcal{P}_{-d} = \{p \text{ prime} \mid (\frac{-d}{p}) = 1\}$. It is easy to determine

Algorithm 1: DL-algorithm in $CL(-d)$

Input: Discriminant $-d$ of an imaginary quadratic field K ,
two form classes $[g], [h] \in CL(-d)$, error probability ϵ

Output: either natural l such that $[g]^l = [h]$ or UNDEF,
meaning that with probability $1 - \epsilon$ there is no such l .

IQDL($-d, g, h$)

1. Construct the factor base \mathcal{F} :
 $\mathcal{F} := \{[f] \mid f = (p, b, \cdot), p \in \mathcal{P}_{-d}, p < L_d(\frac{1}{2}, \frac{1}{\sqrt{8}})\}$
2. Construct the generating set \mathcal{G} :
 $\mathcal{G} := \{[f] \mid f = (p, b, \cdot), p \in \mathcal{P}_{-d}, p < 6 \log^2 d\}$
3. Construct the extended factor base \mathcal{E} :
 $\mathcal{E} := \mathcal{F} \cup \mathcal{G} \cup \{g, h\}$
4. **foreach** $f \in \mathcal{E}$
 $v^{(f)} := \text{IQRELATION}(f, 2nd, \mathcal{G} \cup \{f, g\}, n^2 d)$
5. **for** $i := 1$ **to** $3n \log d - 3 \log \epsilon$
 $v^{(i)} := \text{IQRELATION}(1, 0, \mathcal{E}, d^2)$
6. Collect relations $v^{(i)}$ and $v^{(f)}$ into matrix $A =: (\frac{a}{A'})$ with first row a containing exponents of g
7. DIOPHANTINESOLVER($A', e_1, \epsilon/2$) =: (y, d)
8. **if** $A'y = e_1$ **then return** $l := a \cdot y$
else return UNDEF
- 9.

whether any given p belongs to \mathcal{P}_{-d} . Let $n_0 = \lceil L_d(\frac{1}{2}, \frac{1}{\sqrt{8}}) \rceil$. We collect into \mathcal{F} the first n_0 prime forms (p, b, \cdot) with $p \in \mathcal{P}_{-d}$.

In order to produce random forms we need a set of generators of $Cl(-d)$. Due to a theorem by Bach [Bac90] it suffices to use the set $\mathcal{G} = \{[f] \mid f = (p, b, \cdot), p \text{ prime}, p < c \log^2 d\}$ where c can be chosen to be 6 in the case of quadratic fields, and 12 for general number fields.

Finally, we represent $Cl(-d)$ as a quotient of the lattice spanned by the union $\mathcal{E} = \mathcal{F} \cup \mathcal{G} \cup \{g, h\}$, which we will call the extended factor base. Remember that for large d $\mathcal{G} \subset \mathcal{F}$ so that we really add only g and h to \mathcal{F} . Let $n := \text{card } \mathcal{E} = \max(\lceil L_d(\frac{1}{2}, \frac{1}{\sqrt{8}}) \rceil, 6 \log^2 d) + 2$.

We have the obvious group homomorphism

$$\phi : \mathbb{Z}^{\mathcal{E}} \longrightarrow Cl(-d) : (e_f)_{f \in \mathcal{E}} \longmapsto \prod_{f \in \mathcal{E}} [f]^{e_f}.$$

Its kernel Λ is a sub-lattice of full rank since $Cl(-d)$ is finite. We construct a set \mathcal{H} of $m := n^{1+o(1)}$ relations $v \in \mathbb{Z}^{\mathcal{E}}$ which generate Λ for sufficiently large d with probability $1 - \epsilon/2$, where the error probability for the total algorithm ϵ is given in advance. To this end we follow the procedure of [HM89]: we compute random form classes, and try to factor their reduced representatives over our factor base \mathcal{F} .

Algorithm 2: Generation of relations

Input: form f , exponent u , set of generators \mathcal{H} , radius r

Output: relation $v = (v_e)_{e \in \mathcal{E}}$ s.th. $|v_f - a| < \log d$, and for $e \neq f$ $|v_e| < r + \log d$ if $e \in \mathcal{H}$, or else $|v_e| < \log d$

IQRELATION(f, a, \mathcal{H}, r)

1. **repeat**

2. Draw random $(u_e)_{e \in \mathcal{H}}$ from $\mathbb{N}_{\leq r}^{\mathcal{H}}$ with the uniform distribution
3. Let $f' = (a, b, c)$ be the reduced form in the class $f^u \prod_{e \in \mathcal{H}} e^{u_e}$.
4. **until** attempt to factor a with Algorithm 7.2 out of [LP92] is successful
where we choose $y := L_d(\frac{1}{2}, \frac{1}{\sqrt{8}})$ as upper bound for divisors of a .
5. Find with method (2.8) of [LP92] $(t_e)_{e \in \mathcal{F}}$ s.th.

$$(a, b, c) = \prod_{e \in \mathcal{F}} e^{t_e},$$

and let $t_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{F}$.

6. **return** $(s_e)_{e \in \mathcal{E}}$, where

$$s_e := \begin{cases} u + u_e - t_e & \text{if } e = f, \\ u_e - t_e & \text{if } e \in \mathcal{H}, e \neq f, \\ -t_e & \text{if } e \in \mathcal{E} \setminus \mathcal{H}. \end{cases}$$

In order to generate the first n relations we start as in [Sey87] with large factors f^{2nd} , where f runs through \mathcal{E} , multiply each with random forms from (the image under ϕ of) a box in $\mathbb{Z}^{\mathcal{G} \cup \{f,g\}}$ of radius n^2d until we find a multiple that can be factored by the elliptic curve method. Lenstra and Pomerance [LP92], Theorem 8.1., show that these multiples can be factored with probability larger than $L_d(\frac{1}{2}, -\frac{1}{\sqrt{2}})$; note that its preconditions are fulfilled since we assume the GRH (cf. the Remark following the proof of Theorem 8.1.). The n relations found generate already a full-ranked sub-lattice Λ_0 of Λ because the relation vectors can be arranged to form a diagonally dominant matrix. In contrast to [HM89], we do not need to compute the determinant of Λ_0 .

The rest of our relations are then chosen as in [HM89] to be approximately evenly distributed in Λ/Λ_0 by drawing random forms from (the image of) a box in $\mathbb{Z}^{\mathcal{E}}$ of radius d^2 . We do not check whether \mathcal{H} indeed generates Λ since by adjusting m we can achieve that it will do so with predetermined probability $\epsilon/2$. (cf. [HM89]. If d has more than 10 decimal digits $3n \log d - 3 \log(\epsilon/2)$ will suffice.)

Assume for the moment that \mathcal{H} indeed generates Λ . Let A be the matrix whose column vectors are the $v \in \mathcal{H}$. We may arrange the rows of these vectors in such a way that the entries corresponding to the exponents of g and h appear in the first and second row, respectively. Then the DL problem is solvable if and

only if Λ contains some vector of the form $(l, 1, 0, \dots, 0)^T$. Due to our assumption that \mathcal{H} generates Λ this happens in turn iff

$$A'y = (1, 0, \dots, 0)^T \quad (1)$$

is solvable, where A' is obtained from A by striking out the first row a . If y is a solution of the Diophantine linear system (1), then we have $[g]^l = [h]$ for $l = a \cdot y$ since $(l, 1, 0, \dots, 0)^T \in \Lambda$. Note that the value l thus found is not necessarily minimal.

The algorithm DIOPHANTINESOLVER we apply for the solution of (1) was recently developed by Mulders and Storjohann [MS99]. Given as input a system $Ay = b$ that is solvable over \mathbb{Q} , and some error probability δ DIOPHANTINE-SOLVER yields with probability $1 - \delta$ a pair (y, d) with natural d , and integral y whose entries are bounded in bit length by $O^\sim(n)$ such that

$$Ay = db, \quad (2)$$

and the output d is minimal among all pairs fulfilling (2). The remaining cases which occur with probability $\delta/2$ each are that d is not minimal, or no solution is given at all. The algorithm is probabilistic, and even when successful may return different y in different runs. At any rate, (1) is solvable if a pair (y, d) with $d = 1$ is returned.

There remains the unlikely case that we have not found a complete lattice of relations. In this case (1) will not necessarily be solvable, even though the DL problem might be. Thus if the check in the last step of Algorithm 1 is negative (the case labeled UNDEF in the listing of the algorithm) then we know that either (a) $[h]$ does not lie in $\langle [g] \rangle$, or—with controlled, small probability—(b) \mathcal{H} does not generate Λ/Λ_0 , or (c) DIOPHANTINESOLVER returned no solution or one with inaccurate denominator. If one is willing to invest some more time (within the same asymptotic time constraints) then it is possible to certify that indeed case (a) precludes finding a relation $(l, 1, 0, \dots, 0)$. We will deal with this task in section 4.

3 Running Time Analysis

Steps 1 to 3 take at most $O(n)$ bit operations (cf. [McC89]).

For an estimate of the running time of steps 4 and 5 we apply the analysis in [LP92]. The collection of $n^{1+o(1)}$ relations takes expected time $n^{1+o(1)}$. $L_d(\frac{1}{2}, \frac{1}{\sqrt{2}}) = L_d(\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1))$ multiplied by the time needed for the actual factorization which can be absorbed into the $o(1)$ term. This term is effectively computable on the basis of the data given in [LP92], and our estimate of m .

The solution of the system in step 9 uses an expected number of $O(n^{3+o(1)}) = L_d(\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1))$ bit operations. For the analysis of the perturbation method see [MS99]. The final step of our algorithm takes time $O(n)$.

In consequence we have the following

Theorem 1. (GRH) *There exists a probabilistic algorithm that decides for discriminant $-d$, and forms $g, h \in Cl(-d)$ with error probability ϵ given in advance and independent of d, g, h whether there exists an l such that $g^l \sim h$, and computes some l in expected time $L_d(\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1))$.*

4 Non-solvability of the DL Problem and the Computation of the Class Group

There remains the blemish on Algorithm 1 that it is not able to certify that there is no solution of the DL problem. In order to be able to do this we have to verify that the set of relations found in Algorithm 1 generates the full relation matrix. As a side product we compute the class number, and may also optionally extract the structure of the class group of the given imaginary quadratic field. Once we have established that we have the full relation lattice we will use the algorithm proposed in [GLS98] to certify inconsistency of the diophantine system (1) over \mathbb{Z} .

Algorithm 3: Verification of lattice of relations

Input: discriminant $-d$ of imaginary quadratic field K ;
relation matrix A of some (extended) factor base \mathcal{E} containing generating set \mathcal{G} ; the first $\text{card}\mathcal{G}$ rows of B are occupied by entries corresponding to exponents of elements of \mathcal{G}
Output: TRUE if the columns of B generate a full relation matrix of \mathcal{E} ,
ERROR else

1. Using the class number formula compute a rational number \tilde{h} with $\tilde{h}/2 \leq h(-d) < \tilde{h}$
 2. Compute $C := \text{TRIANG}(A, \text{card}\mathcal{G}, \epsilon)$. (With probability $1 - \epsilon$ an upper triangular C will be returned which will generate the full relation lattice of \mathcal{G}).
 3. **if** TRIANG returned ERROR or $\det C$ does not lie in the range predicted by step 1 **then return** ERROR
 4. **return** TRUE.
 5. [Optional] Compute the Smith normal form of C to obtain the structure of $Cl(-d)$.
-

Let Λ' be the lattice generated by \mathcal{H} . In order to check whether Λ' coincides with the full relation lattice Λ we calculate the determinant of Λ' as the determinant of the (essential part) of the HNF of a matrix B whose columns are the vectors in \mathcal{H} . This time we arrange the rows in such a way that the exponents of elements in \mathcal{G} come first, and enumerate $\mathcal{G} =: \{g_1, \dots, g_q\}$ according to the row numbers in B . Obviously, the essential part of the HNF of B is restricted to these rows.

Algorithm 4: Triangularization of matrix with small essential part of HNF

Input: $n \times m$ matrix B of full rank; r highest number of a row in the HNF of B with diagonal entry not 1; error probability ϵ

Output: triangular C_0 with $C := \begin{pmatrix} 0 & C_0 & * \\ 0 & 0 & I \end{pmatrix} = BU$ for some $U \in Gl(m, \mathbb{Z})$

TRIANG(B, r, ϵ)

1. For $k := n$ to $n - r + 1$, let $B^{(k)}$ be the $k \times m$ matrix obtained from B by striking out the first $n - k$ rows. Let $e_1^{(k)}$ be the k -dimensional column vector $(1, 0, \dots, 0)^T$.
 2. Let $(y^{(k)}, d^{(k)}) := \text{DIOPHANTINE SOLVER}((B^{(k)}, e_1^{(k)}, \epsilon/r^2))$. Collect the m -dimensional column vectors $y^{(k)}$ into the matrix Y .
 3. **if** none of the calls to DIOPHANTINE SOLVER returned ERROR **then return** C_0 , the matrix obtained by striking out all but the first r rows of the product AY .
-

Let C be the HNF of B with zero columns removed. Let further $(c_k)_{k=1}^n$ be the diagonal elements of C . Then

$$c_k = \min\{c \mid (\underbrace{*}, \dots, *, c, 0, \dots, 0)^T \in \Lambda'\}. \quad (3)$$

$(k-1)$ -times

(If $\Lambda' = \Lambda$ then we have also $c_k = \min\{c \mid g_k^c \in \langle g_1, \dots, g_{k-1} \rangle\}$.) The c_k are easily found with the methods that were already employed in section 2: Strike out the corresponding number of rows of B which will yield some $B^{(k)}$, and try to solve

$$B^{(k)} y^{(k)} = e_1^{(k)}, \quad (4)$$

where $e_1^{(k)} = (1, 0, \dots, 0)^T$ of the necessary dimension. From DIOPHANTINE-SOLVER we will get with probability $1 - \delta/2$ some pair $(y^{(k)}, d_k)$ where d_k with conditional probability $1 - \delta/2$ is minimal, and, thus, equals c_k . This calculation needs to be done only $q = 6 \log^2 d$ times, since \mathcal{G} generates the class group, and, hence, the remaining c_k for $k > q$ are 1.

It remains for us to collect the $(y^{(k)})_{k=1}^q$ into a transformation matrix Y , multiply B with Y , and read off the d_k . Their product is the sought determinant h' of Λ' . By comparing h' with bounds for $h := h(-d) = \det \Lambda$ which we may obtain from the analytic class number formula as e.g. in [McC89] we can now decide whether $\Lambda' = \Lambda$.

In order to limit the error probability of the verification algorithm to our ϵ we need to adjust appropriately the error probability δ we set for each call to DIOPHANTINE-SOLVER. A rough, but sufficient estimate would be $\delta := \epsilon/q^2$ where $q = \text{card } \mathcal{G} = 6 \log^2 d$ as above. This has little effect on the total running time of the algorithm since the complexity of DIOPHANTINE-SOLVER grows logarithmically with δ . The total time for the verification of $\Lambda' = \Lambda$ is hence $q \cdot O(\log \log d) \cdot O(n^{3+o(1)})$ which is still $O(n^{3+o(1)})$.

Considering that the calculation of the Smith normal form of a matrix of size q takes at most $O(q^4)$ operations we have as a by-product the following

Theorem 2. (GRH) *There exists a Las-Vegas algorithm that computes the class number, and the structure of the class group of an imaginary quadratic field with discriminant $-d$ with error probability ϵ given in advance and independent of d in expected time $L_d(\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1))$.*

Once we are assured that (1) does have a solution whenever the DL-problem is solvable we turn to the certification algorithm by Giesbrecht, Lobo, and Saunders [GLS98]. Their algorithm CERTIFYZINCONSISTENCY needs square matrices with rational solutions as input. By replacing \mathcal{G} with $\mathcal{G} \cup \{g\}$ in the above triangularization process we obtain a suitable small $(q+1) \times (q+1)$ -matrix C . This matrix is triangular, and there is an (easily computable) triangular unimodular matrix U such that CU is in Hermite normal form. This is also the essential part of the HNF of A' which can be written as

$$H := \begin{pmatrix} 0 & CU & 0 \\ 0 & 0 & I \end{pmatrix}.$$

Thus, a solution of $Cy = e_1$ would yield one for (1) which we assume not to exist. Furthermore, a certificate for non-solvability of $Cy = e_1$ can easily be translated into one for (1). Indeed, CERTIFYZINCONSISTENCY yields a prime p and a row vector u s.th. $uC = 0 \pmod{p}$, but $u \cdot e_1 = u_1 \neq 0 \pmod{p}$. Extending u with zeroes to dimension $n-1$ yields the vector u' which annihilates H , and therefore also A' . Since its first entry is of course still nonzero we have $u'A = 0 \pmod{p}$, but $u' \cdot e_1 \neq 0 \pmod{p}$, i.e. u' is a certificate for non-solvability of (1).

Summarizing, we have found an algorithm for certifying the non-solvability of the DL-problem.

Algorithm 5: Certify non-existence of DL in $Cl(-d)$

Input: discriminant $-d$ of an imaginary quadratic field K ;

$n \times m$ matrix A of relations between elements of an extended factor base, the first $q+2$ rows of which contain the entries corresponding to argument h and base g of the logarithm, followed by the members of a generating set \mathcal{G} .

We assume there is no y s.th. $Ay = (*, 1, 0, \dots, 0)^T$
error probability ϵ

Output: $n \times 1$ -vector u with nonzero second entry such that $uA = 0$

1. Remove the first row from A to obtain B , and set $C = \text{TRIANG}(B, q+1, \epsilon)$
 2. Let $u = \text{CERTIFYZINCONSISTENCY}(C, e_1, \epsilon)$, and let $u' = (0, u_1, \dots, u_{q+1}, 0, \dots, 0)$
 3. **if** $u'A = 0$ **then return** u'
-

This algorithm may already be applied when Algorithm 1 returns UNDEF. In this case we need to limit the time spent on CERTIFYZINCONSISTENCY, and pronounce failure whenever no certificate is returned in this time. With this approach we have an algorithm of Las Vegas type that decides the DL-problem, and whose output can rapidly be checked whether the DL-problem is solvable or not.

5 The DL-Problem in the Real Quadratic Case

The real quadratic case differs from the imaginary one by three effects: (a) class groups of real quadratic fields are usually small, (b) real quadratic fields have non-trivial units, and (c) the notion of a reduced form, and the algorithm of reducing a given form differ somewhat from the imaginary quadratic case: Most significantly, there is not one reduced form in each class, but a cycle of forms which is traversed by successive reduction.

Factor (a) leads us to use the ideal group instead of the class group as the setting for our problem.

The RQNF-DL problem. Let K be a real quadratic field of discriminant d . Given two ideals \mathfrak{g} and \mathfrak{h} in I_K the task is to decide whether there exists an exponent $l \in \mathbb{Z}$, and some $\alpha \in K$ such that

$$\mathfrak{g}^l = \alpha \cdot \mathfrak{h},$$

and, if the answer is positive, to find one pair (l, α) .

For ease of calculation, we will continue to work with binary forms as well as ideals. Thus, we will switch back and forth between representing the class group by ideals or forms. Thus $Cl(d)$, the ideal class group, is identified with the set of $PSL_2(\mathbb{Z})$ -equivalence classes of indefinite primitive binary quadratic forms of discriminant d modulo relations $(a, b, c) - (-a, b, -c)$, which is endowed with the group structure induced by Gaussian composition. As before, the class with representative f will be denoted by $[f]$. Additionally, we adopt the notation used in Chapter 5 of [Coh93], and will denote the map from forms to ideals, or ideals to forms, and their corresponding classes by $\phi_{FI}, \phi_{IF}, \psi_{FI}, \psi_{IF}$, respectively.

Factor (c) forces us to adapt our algorithm: we need to specify the number of reductions which are to be performed at each composition, and a method that ensures the (approximately uniform) selection of a reduced representative in each form class. The first is easily done by requiring the minimal number of reductions that is needed to arrive at a reduced form. The latter is achieved by the algorithm REACH proposed by Abel in [Abe94]. (Given as input some uniformly distributed y from a region in \mathbb{R} that is sufficiently large in comparison with the regulator, and some reduced ideal, her algorithm yields in polynomial time a random reduced ideal in the same class, and a relative generator of the pair which is of size close to y .)

Algorithm 6: DL-algorithm in $CL(d)$

Input: Discriminant d of a real quadratic field K ,
two ideals $\mathfrak{g}, \mathfrak{h} \in I_K$, error probability ϵ

Output: either natural l such that $\mathfrak{g}^l = \mathfrak{h}$, some $\mathcal{B} \subset K$, and a vector
of natural numbers $(s_\beta)_{\beta \in \mathcal{B}}$ each of bit length bounded by
 $O(n)$, where $n := \max(L_d(\frac{1}{2}, \frac{1}{\sqrt{8}}), 6 \log^2(d)) + 2$ such that

$$\phi_{FI}(g)^l = \left(\prod_{\beta \in \mathcal{B}} \beta^{s_\beta} \right) \cdot \phi_{FI}(h)$$

or UNDEF, meaning that with probability $1 - \epsilon$ there is no
such l .

RQDL($d, \mathfrak{g}, \mathfrak{h}$)

1. Construct the factor base \mathcal{F} :
 $\mathcal{F} := \{[f] \mid f = (p, b, \cdot), (\frac{d}{p}) = 1, p < L_d(\frac{1}{2}, \frac{1}{\sqrt{8}})\}$
 2. Construct the generating set \mathcal{G} :
 $\mathcal{G} := \{[f] \mid f = (p, b, \cdot), (\frac{d}{p}) = 1, p < c \log^2 d\}$
 3. Construct the extended factor base \mathcal{E} :
 $\mathcal{E} := \mathcal{F} \cup \mathcal{G} \cup \{\phi_{IF}(\mathfrak{g}), \phi_{IF}(\mathfrak{h})\}$
 4. **foreach** $f \in \mathcal{E}$
 $(v^{(f)}, \mathcal{B}_f, (s_\beta^{(f)})_{\beta \in \mathcal{B}_f}) := \text{RQRELATION}(f, 2nd, \mathcal{G} \cup \{f, g\}, n^2 d)$
 5. **for** $i := 1$ **to** $3n \log d - 3 \log \epsilon =: m_0$
 $(v^{(i)}, \mathcal{B}_i, (s_\beta^{(i)})_{\beta \in \mathcal{B}_i}) := \text{RQRELATION}(1, 0, \mathcal{F}, d^2)$
 6. Collect relations $v^{(i)}$ and $v^{(f)}$ into matrix $A =: (\frac{a}{A'})$ with first
row a containing exponents of h
 7. DIOPHANTINESOLVER($A', e_1, \epsilon/2$) =: (y, d)
 8. **if** $A'y \neq e_1$ **then return** UNDEF
 9. Let $l = a \cdot y$
 10. Let $\mathcal{B} := \bigcup_{f \in \mathcal{E}} \mathcal{B}_f \cup \bigcup_{i=1}^{m_0} \mathcal{B}_i$; reindex the $s^{(f)}$, and $s^{(i)}$ with
index set \mathcal{B} , and set $s_\beta := \sum_{f \in \mathcal{E}} y_f s_\beta^{(f)} + \sum_{i=1}^{m_0} y_i s_\beta^{(i)}$
 11. **return** $(l, \mathcal{B}, (s_\beta)_{\beta \in \mathcal{B}})$
-

Furthermore, we will modify our algorithm to yield the additional information necessary to calculate the relative generator α . Each time we arrive at a relation v we record its generator α_v , i.e. the generator of the ideal $\prod_e \phi_{FI}(e)^{v_e}$. The vector $y = (y_v)$ giving a combination of relations that equals a “DL relation” $(*, 1, 0, \dots, 0)$ can now also be interpreted as an exponent vector for calculating $\alpha := \prod_v \alpha_v^{y_v}$.

We will not actually calculate the α_v and α explicitly since this would take exponential time and space to complete. These quantities will be expressed in compact form as pairs of two vectors containing bases and exponents, respectively. Note, however, that in practical cryptographic circumstances α will have a short representation which we are able to compute efficiently on the basis of this vector pair by an approximation technique.

We will give the “real” versions of Algorithm IQDL, and IQRELATION on pages 590, and 592 respectively.

When reading algorithm RQDL it is best to imagine that the matrix composed in step 6 indeed contains one more row in which the generators of the relation are recorded:

$$\tilde{A} := \begin{pmatrix} \vdots & \vdots & \vdots \\ \cdots v_e^{(f)} & \cdots v_e^{(1)} & \cdots v_e^{(m_0)} \\ \vdots & \vdots & \vdots \\ \cdots \alpha_f & \cdots \alpha_1 & \cdots \alpha_{m_0} \end{pmatrix}$$

If the call to DIOPHANTINESOLVER in step 7 returns a solution $(y, 1)$ then $\tilde{A} \cdot y = (l, 1, 0, \dots, 0, \alpha)^T$, where operations in the last row are in the multiplicative group K^* .

We must convince ourselves that the α_* which are returned in compact form by RQRELATION do indeed generate the principal ideal associated with the relation generated. For this one needs to follow the composition of forms in step 8. Each time, the reduction operator is applied on a form $f = (a, b, c)$ we gather as factor the relative generator $\beta = (b + \sqrt(d))/2a$ for which $\phi_{FI}(\rho(f)) = \beta\phi_{FI}(f)$. Doing this successively yields

$$\begin{aligned} \prod \phi_{FI}(e)^{v_e} &= \prod \beta_* \cdot \phi_{FI}(\prod e^{v_e}) \\ &= \prod \beta_* \cdot \mathbf{1}. \end{aligned} \tag{5}$$

The last equality holds since the product on the right is equivalent to $\mathbf{1}$ not only modulo $\text{PSL}_2(\mathbb{Z})$ but by construction already modulo Γ_∞ , the kernel of ϕ_{FI} .

The time needed for the execution of the real quadratic algorithms does not differ asymptotically from those of their imaginary quadratic counterparts despite the additional bookkeeping involved. (There is of course a slight change in the $o(1)$ term.) Note in particular that RQRELATION generates in steps 2 to 4 random reduced forms with a sufficiently uniform distribution, and that the probability that random reduced forms can be factored does not differ between the positive definite, and the indefinite case.

In consequence we have the following

Theorem 3. (GRH) *There exists a probabilistic algorithm that decides for discriminant d , and forms $\mathfrak{g}, \mathfrak{h} \in Cl(-d)$ with error probability ϵ given in advance and independent of $d, \mathfrak{g}, \mathfrak{h}$ whether there exists α, l such that $\mathfrak{g}^l = \alpha \cdot \mathfrak{h}$, and computes some l , and some α in expected time $L_d(\frac{1}{2}, \frac{3\sqrt{2}}{4} + o(1))$.*

Algorithm 7: Generation of relations among generators of $Cl(d)$

Input: form f , exponent u , set of generators \mathcal{H} , radius r

Output: relation v s.th. $|v_f - u| < \log d$, and for $e \neq f$ $|v_e| < r + \log d$
 if $e \in \mathcal{H}$, or else $|v_e| < \log d$
 set $\mathcal{B} \subset K$, and vector $(s_\beta)_{\beta \in \mathcal{B}}$ s.th. $\prod_{e \in \mathcal{E}} e^{v_e} = \left(\prod_{\beta \in \mathcal{B}} \beta^{s_\beta} \right)$

RQRELATION(f, u, \mathcal{H}, r)

1. **repeat**
2. Draw random $(u_e)_{e \in \mathcal{H}}$ from $\mathbb{N}_{<r}^{\mathcal{H}}$ with the uniform distribution
3. Let $f' := f^u \prod_{e \in \mathcal{H}} e^{u_e}$. (In the course of the composition of two forms, the reduction operator ρ is applied the minimal number of times to reach a reduced form.)
4. Select a random reduced form $f'' = (a'', b'', c'')$ in the cycle of f' by choosing some $y \in [-d, 0]$, and, employing REACH as defined in [Abe94] to find γ , and f'' s.th. $\ln|\gamma| \approx y$, and $\phi_{FI}(f'') = (\gamma) \cdot \phi_{FI}(f')$.
5. **until** attempt to factor a'' with Algorithm 7.2 out of [LP92] is successful where we choose $y := L_d(\frac{1}{2}, \frac{1}{\sqrt{8}})$ as upper bound for the divisors of a'' .
6. For all $e \in \mathcal{F}, e = (p, b_p, \cdot)$ let t_e be s.th. $p^{|t_e|} \parallel a''$, and $b \equiv \text{sign}(t_e)b_p \pmod{2p}$. Thus

$$(a'', b'', c'') = \prod_{e \in \mathcal{F}} e^{t_e}.$$

Let further $t_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{F}$.

7. Compute the exponents $(v_e)_{e \in \mathcal{E}}$, where

$$v_e := \begin{cases} u + u_e - t_e & \text{if } e = f, \\ u_e - t_e & \text{if } e \in \mathcal{H}, e \neq f, \\ -t_e & \text{if } e \in \mathcal{E} \setminus \mathcal{H}. \end{cases}$$

8. Recompute $\prod_{e \in \mathcal{E}} e^{v_e}$ this time keeping track of the factors introduced by the reduction steps along the route: for every pair of forms e_1, e_2 to be composed we obtain β_{e_1, e_2} s.th.

$$\phi_{FI}(e_1) \cdot \phi_{FI}(e_2) = \beta(e_1, e_2) \cdot \phi_{FI}(e_1 \cdot e_2).$$

Collect the β found together with γ from step 4 into \mathcal{B} counting multiplicities with the exponents s_β .

9. **return** $(v_e, \mathcal{B}, (s_\beta)_{\beta \in \mathcal{B}})$
-

Contrary to the imaginary quadratic case, we are not able yet to certify within the same time constraints that some \mathfrak{h} does not lie in the subgroup $\langle \mathfrak{g} \rangle \subset CL(d)$. In order to verify whether the lattice spanned by the v_* found is a complete relation lattice we need a bound on the class number. Here, the analytic class number formula delivers only bounds on the product of class number and regulator. Thus we would need to compute the regulator first. To the best of our knowledge, the algorithms currently available to do this on the basis of the data already accumulated are too slow, with time constrained in our set-up by $L_d(\frac{1}{2}, \sqrt{2} + o(1))$.

[One such algorithm would be a speed up of Buchmann's algorithm [Buc89] combining ideas of Abel [Abe94] with a recently developed algorithm by Maurer [Mau00] that enables us to rapidly calculate a fundamental unit given a generating set of units. In [Abe94], Abel shows following closely [Buc89], and [BK92] how to generate sufficiently many relations so that the logarithms of their relative generators span a lattice in \mathbb{R} that contains the fundamental unit, and how to control the logarithms of these relative generators in the course of the regulator computation.]

However, it seems very likely that a further, minor speed-up of the linear algebra step of the outlined algorithm should allow to sharpen the time estimate to match that of the imaginary quadratic case.]

6 Conclusion

We have analysed the DL-problem in two environments for which there have been proposed cryptographic (e.g. key exchange) protocols based on the difficulty of this problem. The asymptotic behaviour of the algorithms presented here though considerably better than that of previous algorithms lags still behind the best rigorously analysed one for the similar problem in the multiplicative group of finite fields, or those for factoring integers. This may indicate that algorithms which work in groups whose size is difficult to compute are harder to break than those in groups of known size.

It remains a very interesting problem whether this gap can be closed, i.e. whether there are algorithms which are amenable to rigorous analysis which compute the DL in class groups in time $L_d(\frac{1}{2}, 1 + o(1))$. Moreover, it would be very interesting to see whether methods similar to the general number field sieve can be applied in this environment in order to find an algorithm that at least heuristically matches the state of the art in factoring integers. However, if that turns out to be beyond the reach of current methods, and ideas, we would have a strong indication that algorithms in class groups would indeed deliver enhanced security over RSA-type algorithms.

There also remain more straightforward extensions of the current work. We would just like to mention on the easy side the calculation of the regulator of real quadratic fields, and the extension to non-maximal quadratic orders, on the more difficult one the removal of the assumption that the Generalised Riemann Hypothesis holds, and the extension to class groups of fields of higher degree.

References

- [Abe94] C.S. Abel. *Ein Algorithmus zur Berechnung der Klassenzahl und des Regulators reellquadratischer Ordnungen*. PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1994.
- [Bac90] E. Bach. Explicit bounds for primality testing and related problems. *Math. Comp.*, 55:355–380, 1990.
- [BD90] J. Buchmann and S. Düllmann. On the computation of discrete logarithms in class groups. In *Proc. of CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 134–139. Springer, 1990.
- [BK92] J. Buchmann and V. Kessler. Computing a reduced lattice basis from a generating system. Unpublished manuscript, 1992.
- [Buc89] J. Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In *Séminaire de Théorie des Nombres*, pages 27–41, Paris, 1988–89.
- [BW88] J. Buchmann and H.C. Williams. A key-exchange system based on imaginary quadratic fields. *Journal of Cryptology*, 1:107–118, 1988.
- [Coh93] H. Cohen. *A course in computational algebraic number theory*. Springer, Heidelberg, 1993.
- [GLS98] M. Giesbrecht, A. Lobo, and B.D. Saunders. Certifying inconsistency of sparse linear systems. In O. Gloor, editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '98*, pages 113–119, 1998.
- [HM89] J. L. Hafner and K. S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2:839–850, 1989.
- [LP92] H.W. Lenstra Jr. and C. Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5:483–516, 1992.
- [Mau00] M. Maurer. *Regulator approximation and fundamental unit computation for real quadratic orders*. PhD thesis, TU Darmstadt, 2000.
- [McC89] K. McCurley. Cryptographic key distribution and computation in class groups. In R.A. Mollin, editor, *Number Theory and Applications*, pages 459–479. Kluwer Academic Publishers, 1989.
- [MS99] Th. Mulders and A. Storjohann. Diophantine linear system solving. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '99*, 1999. to appear.
- [Pom87] C. Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In *Discrete Logarithms and complexity, Proc. of the Japan-US joint seminar on discrete logarithms and complexity theory*. Academic Press, 1987.
- [Sey87] M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Math. Comp.*, 48:757–780, 1987.

Asymptotically Fast GCD Computation in $\mathbb{Z}[i]$

André Weilert

Institut für Informatik II
Römerstraße 164, D-53117 Bonn, Germany
weilert@cs.uni-bonn.de

Abstract. We present an asymptotically fast algorithm for the computation of the greatest common divisor (GCD) of two Gaussian integers. Our algorithm is based on a controlled Euclidean descent in that the operands are not reduced too much in each Euclidean step. To compute a descent of n bits, the algorithm recursively calculates two descents of approximately $n/2$ bits each with short operands and transfers the thereby calculated cofactors to the original operands. Overall, this algorithm achieves a time bound of $O(n(\log n)^2 \log \log n)$ bit operations for operands bounded by 2^n in absolute value.

Furthermore, we show that the biquadratic residue symbol of two bounded Gaussian integers can be computed as fast as the GCD can be calculated (except for a larger constant hidden in the O -notation). This result is achieved by first calculating the GCD of the operands, and then using the sequence of quotients from the Euclidean descent to compute the biquadratic residue symbol.

1 Introduction

This paper deals with an asymptotically fast algorithm for computing the greatest common divisor (GCD) of two Gaussian integers, i. e. complex numbers with real and imaginary part both integer. The main idea is that one does not operate with the whole operands all the time, but instead uses prefixes of the operands to calculate a Euclidean descent, i.e., a sequence of Euclidean steps where the remainders are greater than a size bound. This technique is adapted from GCD calculation in \mathbb{Z} ; it is called controlled Euclidean descent, first used by Schönhage [Sch87] for fast GCD calculation in \mathbb{Z} . We will discuss our new GCD algorithm for $\mathbb{Z}[i]$ in detail and will prove that it achieves a time bound of $O(n(\log n)^2 \log \log n)$ bit operations for Gaussian integers bounded by 2^n in absolute value. After this we give a brief overview how to calculate the biquadratic residue symbol in a fast manner. This is an application of the fast GCD calculation in $\mathbb{Z}[i]$. Finally, we look at some practical running time of several GCD algorithms in $\mathbb{Z}[i]$.

First we review previous work on GCD algorithms in \mathbb{Z} and $\mathbb{Z}[i]$. We see that some of the techniques used to accelerate the GCD calculation in \mathbb{Z} can be used to accelerate the GCD calculation in $\mathbb{Z}[i]$. Thus it is helpful to also discuss the development of GCD algorithms in \mathbb{Z} .

Since Euclid (cf. [Hea56, Book VII, Propositions 1–3]) it is well-known how to calculate a GCD in \mathbb{Z} in polynomial time. However, the computation with the so-called Euclidean algorithm is not optimal. Lehmer [Leh38] developed a technique for faster GCD calculation that uses as long as possible only single-precision prefixes of the operands in the Euclidean steps. With every Euclidean step, the cofactors are also updated such that the calculated reduction can be transferred to the whole operands. This algorithm speeds up the standard Euclidean algorithm because the rounded quotient of two random integers will be less than 1000 approximately 99.856 percent of the time, cf. Knuth [Knu98, 4.5.2, high-precision calculation]. Therefore it suffices to calculate most of the Euclidean steps in single-precision. Sorenson [Sor95] refined this algorithm and achieved a running time of $O(n^2 / \log n)$. Due to Stein [Ste67] is the so-called binary algorithm. This algorithm uses another idea for speeding up the GCD calculation in \mathbb{Z} ; it is based on the fact that the difference of two odd integers is even and thus divisible by two. At first, the operands are made odd by dividing them by two as long as possible. Then in every single reduction, one calculates the difference of the odd intermediate operands and replaces the larger intermediate operand with this difference divided by the maximal power of two. It follows that the intermediate operands are both odd again. It can easily be shown that this reduction makes progress such that the odd part of the GCD will be calculated in this way. The advantage of this algorithm is that it is based on arithmetical operations (addition/subtraction, division by 2) which can easily be done on binary computers. For operands bounded by 2^n in absolute value, these three algorithms have a running time of $O(n^2)$ (cf. [BS96, Chapter 4]), and in case of the binary algorithm, the constant hidden in the O -notation is smaller than the constant in case of the Lehmer-type or Euclidean algorithm. A sophisticated analysis of the running time of the binary GCD algorithm is presented by Knuth [Knu98, Section 4.5.2], but it was already published in the first edition of his book in 1969. Additionally, Brent [Bre76] has discovered a continuous model for some details of Knuth's analysis. Vallée [Val98] has worked out a complete average-case analysis of the binary GCD algorithm. Many people have improved and extended the binary GCD algorithm, e.g., [Nor85, Nor89, Sor90, Jeb93, Sor94, SS94a, SS94b, Web95, SL97], referred to as right-shift, left-shift, or k -ary GCD algorithm, but they have not achieved a better asymptotic time bound than $O(n^2 / \log n)$. Schönhage [Sch71] proved that integer GCD's can be calculated in $O(\mu(n) \log n)$ bit operations using the concept of continued fractions, where $\mu(n)$ denotes an upper bound for the multiplication time of n bit integers on a multitape Turing machine. Schönhage und Strassen [SS71] showed that $\mu(n) \leq O(n \log n \log \log n)$ and such a fast multiplication is already available in software today, see, e.g. [SGV94, 1.3.6, 6.1.3]. Yet this result for the GCD computation was not useful in practice, and for that reason Schönhage [Sch87] developed the *controlled Euclidean descent* for asymptotically fast GCD calculation. With this technique he was able to present a GCD algorithm that calculates the GCD of two n bit integers in time $O(\mu(n) \log n)$,

see [SGV94]. Moreover, he adapted this technique to compute fast reductions of binary quadratic forms [Sch91].

$\mathbb{Z}[i]$ is a Euclidean domain with respect to the absolute value $|\cdot|$. Caviness and Collins [Cav73, CC76] transferred Lehmer's idea to the ring of Gaussian integers. Both algorithms achieve a running time of $O(\mu(n) \log n)$, and this running time bound is larger as in the case of the Euclidean and Lehmer-type GCD algorithm in \mathbb{Z} because the exact calculation of a least remainder in $\mathbb{Z}[i]$ seems to be as expensive as multiplication even if the calculated quotient is small. Another variant to compute the GCD is an $(1+i)$ -adic GCD algorithm [Wei00] that is related to the binary GCD algorithm. It requires no multiplications and only divisions by powers of $1+i$. The prime element $1+i$ plays the rôle that 2 has in the binary GCD algorithm. This $(1+i)$ -adic GCD algorithm achieves a time bound of $O(n^2)$. Thus the standard Euclidean, the Lehmer-type and the binary algorithm are transferred from \mathbb{Z} to analogue algorithms in $\mathbb{Z}[i]$.

Rolletschek [Rol86, Rol89, Rol90] considered Euclidean descents in $\mathbb{Z}[i]$ from a more theoretical point of view and showed that such a descent with each Euclidean step as least remainder leads to the shortest remainder sequence, i.e., the number of Euclidean steps is larger if one calculates at least one Euclidean step with no least remainder. This result holds for four of the Euclidean rings of algebraic integers of the imaginary quadratic number fields $\mathbb{Q}(\sqrt{d})$, $d \in \{-1, -2, -3, -7\}$, but not in the case $d = -11$ which is the fifth Euclidean domain (and these five rings are the only Euclidean domains with respect to $|\cdot|$ of algebraic integers of imaginary quadratic number fields). Kaltofen and Rolletschek [KR89] showed in general how to calculate GCDs in quadratic number fields (with class number 1), in particular in these five Euclidean domains, but their result does not lead to a faster algorithm in $\mathbb{Z}[i]$ than the algorithms known already.

Now we transfer the idea of a controlled Euclidean descent to $\mathbb{Z}[i]$ which will yield the fastest GCD algorithm in $\mathbb{Z}[i]$ today known. This is the subject of the next chapter. For omitted details we refer to Weilert [Wei99a].

2 Controlled Euclidean Descent in $\mathbb{Z}[i]$

A controlled Euclidean descent is a sequence of *modified* Euclidean steps, where the operands are reduced with respect to the Euclidean function (in $\mathbb{Z}[i]$ we use the absolute value $|\cdot|$ as a Euclidean function), however the intermediate operands are not less than a given size bound.

First we will define a Euclidean step, what the cofactors are and show how the cofactors are bounded.

Definition 2.1. A (least remainder) Euclidean step for $x_{j-1}, x_j \in \mathbb{Z}[i]$, $|x_{j-1}| \geq |x_j| > 0$ is a calculation (not unique) of $q_j, x_{j+1} \in \mathbb{Z}[i]$, such that

$$x_{j-1} = q_j \cdot x_j + x_{j+1}, \quad \text{and} \quad |x_{j+1}| \leq |x_j|/\sqrt{2}.$$

Here, the factor $1/\sqrt{2}$ is the best possible that can be achieved in general. Moreover, such a reduction can be calculated effectively with a division of x_{j-1} by x_j in $\mathbb{Q}(i)$ and rounding the real and imaginary part to nearest integers. Furthermore, associated with every Euclidean step is an invertible matrix $Q_j \in \mathbb{Z}[i]^{2 \times 2}$:

$$\begin{pmatrix} x_{j-1} \\ x_j \end{pmatrix} = Q_j \cdot \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix}, \quad Q_j = \begin{pmatrix} q_j & 1 \\ 1 & 0 \end{pmatrix}, \quad \det Q_j = -1.$$

Definition 2.2. Set $M_0 := I$ (2×2 identity matrix), and define $M_j = M_{j-1} \cdot Q_j$ for $j \geq 1$.

If we denote the coefficients of M_j as

$$M_j = \begin{pmatrix} u_{j+1} & u_j \\ v_{j+1} & v_j \end{pmatrix}, \quad (1)$$

then we get $u_0 = 0$, $u_1 = 1$, $u_{j+1} = q_j u_j + u_{j-1}$ and $v_0 = 1$, $v_1 = 0$, $v_{j+1} = q_j v_j + v_{j-1}$. Moreover, $\det M_j = (-1)^j$ and

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = M_j \cdot \begin{pmatrix} x_j \\ x_{j+1} \end{pmatrix}.$$

It follows that $x_j = (-1)^j(v_j x_0 - u_j x_1)$, and u_j, v_j are called the *cofactors* of x_j with respect to x_0, x_1 . The matrix M_j is called the *j-th cofactor matrix*.

We calculate a finite Euclidean descent, consisting of k Euclidean steps as explained above, and get $x_{k+1} = 0$.

Proposition 2.1. For $0 \leq j \leq k$, we get the following size bound for the cofactors:

$$\begin{aligned} |v_{j+1}| &\leq (2 + \sqrt{2}) \cdot |x_1/x_j|, \\ |u_{j+1}| &\leq (3 + \sqrt{2}) \cdot |x_0/x_j|. \end{aligned}$$

Proof. [CC76, Lemma 14, Lemma 15] □

The following theorem is the theoretical background of the controlled Euclidean descent in $\mathbb{Z}[i]$ on which our fast GCD algorithm is based.

Theorem 2.1. Let $x, y \in \mathbb{Z}[i]$, $s \in \mathbb{N}$, $s \geq 1$ and $|x|, |y| \geq s$. There exist $u, v \in \mathbb{Z}[i]$ and $M \in \mathbb{Z}[i]^{2 \times 2}$, such that

$$\begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u \\ v \end{pmatrix}, \quad \det M \in \{-1, +1\}, \quad (2)$$

and

$$2 \max(|x|, |y|) > |u|, |v| \geq s > \min(|u+v|, |u-v|). \quad (3)$$

Additionally, the coefficients of the matrix $M = (m_{ij})$ are bounded by

$$|m_{ij}| \leq \left(4 + \frac{5}{\sqrt{2}}\right) \cdot \frac{\max(|x|, |y|)}{s}. \quad (4)$$

Remark. We will give an algorithmically orientated proof of this theorem because we will have to calculate a part of a controlled Euclidean descent in our fast algorithm. Therefore we present a way to compute a single “controlled” Euclidean step in the following proof.

Proof. W.l.o.g. let $|x| \geq |y| \geq s > 0$. We set $u := x$, $v := y$, and $M := I$. If

$$\min(|u - v|, |u + v|) < s, \quad (5)$$

then we have found a representation as in (2), satisfying (3). Otherwise, we have

$$\min(|u - v|, |u + v|) \geq s, \quad (6)$$

and we calculate a least remainder Euclidean step for u, v , i. e., a Euclidean step that minimizes the remainder with respect to $|\cdot|$.

$$u = qv + r, \quad \text{with} \quad |r| \leq |v|/\sqrt{2} < |v| \leq |u|. \quad (7)$$

If $|r| \geq s$, then update M with

$$M := M \cdot \begin{pmatrix} q & 1 \\ 1 & 0 \end{pmatrix}.$$

If we set $u := v$ and $v := r$, we get

$$\begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u' \\ v' \end{pmatrix}, \quad \max(|x|, |y|) > |u'|, |v'| \geq s.$$

If condition (6) is still satisfied for the intermediate operands u and v , then we continue the calculation of such Euclidean steps as in (7). As $\mathbb{Z}[i]$ is a Euclidean domain, the set $\{|z| : z \in \mathbb{Z}[i], |z| < |y|\}$ is finite (cf. Lenstra [Len74] or Lemmermeyer [Lem95]) and contains 0. Therefore after finitely many Euclidean steps, we calculate a remainder r in (7) with $|r| < s$. As we need an intermediate operand that is larger than s (in particular, for a recursive version of a controlled Euclidean descent), we change the last calculated Euclidean step in the following manner: If $|r - v| \geq s$, then set $\varepsilon := -1$. Otherwise we have $|r - v| < s \leq |v|$, and this yields

$$|r + v| = |2v - (v - r)| \geq 2|v| - \underbrace{|v - r|}_{\leq |v|} \geq |v| \geq s.$$

In this case we set $\varepsilon := 1$. From (7) we achieve (see Fig. 1)

$$u = (q - \varepsilon)v + (r + \varepsilon v), \quad \text{hence} \quad |r + \varepsilon v| \geq s, \quad \varepsilon \in \{-1, +1\}.$$

We set $u' := v$, $v' := r + \varepsilon v$ and update M as follows:

$$M := M \cdot \begin{pmatrix} q - \varepsilon & 1 \\ 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u' \\ v' \end{pmatrix}.$$

The size of u', v' is bounded by

$$|u'| = |v|, \quad |v'| = |r + \varepsilon v| \leq \left(1 + \frac{1}{\sqrt{2}}\right) \cdot |v| < 2|v|.$$

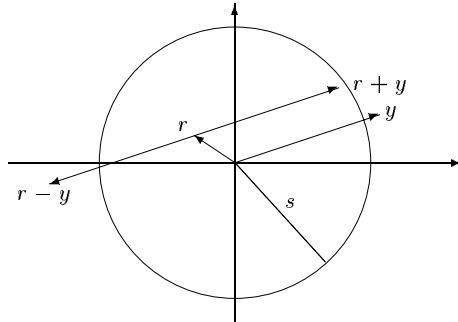
Because of $|u|, |v| < \max(|x|, |y|)$ we get $|u'|, |v'| \leq 2 \max(|x|, |y|)$. Another Euclidean step with operands $u = u'$, $v = v'$ will not be calculated because

$$|u' - \varepsilon^{-1}v'| = |v_{\text{old}} - \varepsilon^{-1}(r + \varepsilon v_{\text{old}})| = |r| < s$$

shows that condition (5) is not satisfied anymore. Thus we have found a representation as in (2), and (3) is fulfilled, too.

We call such a modification of the remainder r to $r + \varepsilon v$ a *size modification after a Euclidean step* (see Fig. 1).

Fig. 1. Size modification after a Euclidean step



Altogether, we calculate k Euclidean steps. Only at the last Euclidean step a size modification can be necessary. For each $1 \leq j \leq k$, we have the invertible matrix

$$\hat{Q}_j = \begin{pmatrix} \hat{q}_j & 1 \\ 1 & 0 \end{pmatrix},$$

where for $j < k$, we have $\hat{q}_j = q_j$, i.e. the quotient calculated by the j -th Euclidean step, and $\hat{q}_k = q_k - \varepsilon$ due to the size modification at the last Euclidean step, $\varepsilon \in \{-1, 0, +1\}$. ($\varepsilon = 0$ iff no size modification was done.) The matrix M is the product of the \hat{Q}_j 's, hence $\det M = (-1)^k$.

To show the size bound of the coefficients of M we use

$$\begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u \\ v \end{pmatrix} = Q_1 \cdot Q_2 \cdot \dots \cdot Q_{k-1} \cdot \hat{Q}_k \begin{pmatrix} u \\ v \end{pmatrix}.$$

Using (1), we get

$$M_{k-1} = Q_1 \cdot \dots \cdot Q_{k-1} = \begin{pmatrix} u_k & u_{k-1} \\ v_k & v_{k-1} \end{pmatrix},$$

and the coefficients of M_{k-1} are size bounded (Proposition 2.1). Thus we have a representation of M as

$$M = M_{k-1} \cdot \hat{Q}_k = \begin{pmatrix} u_k & u_{k-1} \\ v_k & v_{k-1} \end{pmatrix} \cdot \begin{pmatrix} q_k - \varepsilon & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} u_{k+1} - \varepsilon u_k & u_k \\ v_{k+1} - \varepsilon v_k & v_k \end{pmatrix}.$$

Furthermore, we can bound the coefficients of M as follows

$$\begin{aligned} |u_{k+1} - \varepsilon u_k| &\leq |u_{k+1}| + |u_k| < (3 + \sqrt{2}) \cdot \left(\left| \frac{x_0}{x_k} \right| + \left| \frac{x_0}{x_{k-1}} \right| \right) \\ &= (3 + \sqrt{2}) \cdot \left(\left| \frac{x_0}{x_k} \right| + \left| \frac{x_0}{x_k} \cdot \frac{x_k}{x_{k-1}} \right| \right) \leq \left(4 + \frac{5}{\sqrt{2}} \right) \cdot \left| \frac{x_0}{x_k} \right|, \end{aligned}$$

because of $|x_k/x_{k-1}| \leq 1/\sqrt{2}$ for $k \geq 2$. An analogue calculation proves the size bound for $|v_{k+1} - \varepsilon v_k|$. With $|x_k| \geq s$, the claim (4) follows. \square

Proposition 2.2. *After a Euclidean descent from $x, y \in \mathbb{Z}[i]$ to $u, v \in \mathbb{Z}[i]$ as specified in Theorem 2.1 with parameter $s = 1$, we have*

$$u \sim v \sim \gcd(x, y),$$

where \sim denotes equality up to units $\mathbb{Z}[i]^\times$. (Remember that the GCD of two ring elements is unique up to units only. In abuse of notation, we use the functional symbol $\gcd(x, y)$ for an arbitrary generator of the principal ideal $\gcd(x, y)\mathbb{Z}[i]$.)

Proof. We get $u = \varepsilon v$, $\varepsilon \in \{-1, +1\}$, because $|u - \varepsilon v| < s = 1$ yields $u - \varepsilon v = 0$. We have the representation

$$\begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u \\ v \end{pmatrix}. \quad (8)$$

After multiplying this equation with M^{-1} , we get a representation of u as a linear combination of x, y . Therefore, every divisor of x and y is a divisor of u (and v because of $u \sim v$). Assume that $h \cdot \gcd(x, y)$ divides u and v for an $h \in \mathbb{Z}[i]$. From (8), we get representations of x and y as linear combinations of u and v . For that reason, $h \cdot \gcd(x, y)$ is a divisor of x and y . As every divisor of x and y is a divisor of the GCD, it follows that $h \in \mathbb{Z}[i]^\times$ and $u \sim v \sim \gcd(x, y)$. \square

3 The GCD Algorithm

In this section we will present our asymptotically fast GCD algorithm in $\mathbb{Z}[i]$ (Algorithm CIDESCENT), based on the idea of controlled Euclidean descent in $\mathbb{Z}[i]$. We prove that this algorithm is correct and show that its worst-case running time in terms of bit complexity is bounded by $O(\mu(n) \log n)$ for operands less than 2^n in absolute value.

Theorem 3.1. *Algorithm CIDESCENT is correct, i. e., for Gaussian integers $x, y \in \mathbb{Z}[i]$ it calculates $u, v \in \mathbb{Z}[i]$ and a matrix $M \in \mathbb{Z}[i]^{2 \times 2}$ as specified in Theorem 2.1. Furthermore, the number of Euclidean steps in (C5) resp. (C9) is bounded by 2 resp. 5, independent of x, y .*

Algorithm CIDESCENT(x, y, L)

For $x, y \in \mathbb{Z}[i]$, $L \in \mathbb{N}$ with $|x|, |y| \geq 2^L$, **CIDESCENT** calculates Gaussian integers u, v and an invertible matrix $M \in \mathbb{Z}[i]^{2 \times 2}$ (as specified in Theorem 2.1) such that

$$\begin{pmatrix} x \\ y \end{pmatrix} = M \cdot \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{and} \quad |u|, |v| \geq 2^L > \min(|u-v|, |u+v|).$$

C1: if $\min(|x|, |y|) < 2^{L+1}$ then
 $u := x, v := y, M := I;$ $\backslash\backslash$ goto C9

else

C2: Determine the minimal $N \in \mathbb{N}$ such that $|x|, |y| < 2^{L+N};$
if $L \leq N + 5$ then $T := 0, L_1 := L;$ $\backslash\backslash$ no tails

else $L_1 := N + 5, T := L + 1 - L_1,$
split $x = x' + x'' \cdot 2^T,$
 $y = y' + y'' \cdot 2^T,$
such that $2^{L_1} \leq |x''|, |y''| < 2^{L_1+N-1}, |x'|, |y'| \leq 2^{T+\frac{1}{2}},$
and set $x := x'', y := y'';$

C3: $H := L_1 + \lfloor N/2 \rfloor;$
if $\min(|x|, |y|) < 2^H$ then
 $u' := x, v' := y, M := I;$
else

C4: $(u', v', M) := \text{CIDESCENT}(x, y, H);$ $\backslash\backslash \min(|u' - v'|, |u' + v'|) < 2^H$

C5: while $\min(|u' - v'|, |u' + v'|) \geq 2^{L_1}$ and
 $\max(|u'|, |v'|) \geq 2^H$ do $\backslash\backslash$ at most 2 times

Perform one Euclidean step on u', v'
preserving $|u'|, |v'| \geq 2^{L_1},$
and with proper updating of $M;$

if $\min(|u' - v'|, |u' + v'|) < 2^{L_1}$ then
 $u := u', v := v';$
else

C6: $(u, v, M') := \text{CIDESCENT}(u', v', L_1);$ $\backslash\backslash \min(|u - v|, |u + v|) < 2^{L_1}$

C7: $M := M \cdot M'$

C8: if $T > 0$ then
 $u := u \cdot 2^T + \det M \cdot (d \cdot x' - b \cdot y'),$ $\backslash\backslash \det M \in \{-1, +1\}$
 $v := v \cdot 2^T + \det M \cdot (-c \cdot x' + a \cdot y'),$
where $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix};$ $\backslash\backslash M^{-1} = \det M \cdot \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

C9: while $\min(|u - v|, |u + v|) \geq 2^L$ do
Perform one Euclidean step on u, v
preserving $|u|, |v| \geq 2^L,$
and with proper updating of $M;$ $\backslash\backslash$ at most 5 times

C10: return $(u, v, M).$

Proof.

- (i) A size modification after a Euclidean step in the while-loop (C5) (operands u', v') or (C9) (operands u, v) causes the loop to finish because then the minimum condition in the header of the loop is not satisfied anymore.

- (ii) We calculate least remainder Euclidean steps, possibly followed by a size modification. If we calculate at least one Euclidean step after a size modification (possibly done as last Euclidean step in the recursive calls of CIDESCENT) then this step removes the size modification. Consequently we do not have to consider the size modification (apart from the size modification in the last calculated Euclidean step) for bounding the matrix coefficients. It follows that the coefficients are bounded as specified in Theorem 2.1.
- (iii) Assume $|x| \geq |y|$ w.l.o.g. If (C1) branches to (C9), we have $|y| < 2^{L+1}$. Each Euclidean step reduces the remainder – compared with $|y|$ – at least by a factor of $\frac{1}{\sqrt{2}}$. Thus after at most two Euclidean steps, one has calculated a remainder less than 2^L such that a size modification is necessary and after this the loop (C9) will terminate.

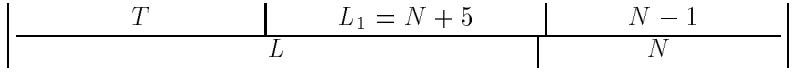
In all other cases, i.e., when (C1) does not branch to (C9), $N \geq 2$ is assured.

- (iv) Now we will show that the number of iterations in (C5) is bounded by 2. In the case $T > 0$, the “heads” x'', y'' of the operands x, y are denoted with \tilde{x}, \tilde{y} . Without loss of generality we assume $|\tilde{x}| \geq |\tilde{y}|$. Otherwise, i.e. when $T = 0$, set $\tilde{x} := x$, $\tilde{y} := y$. We calculate recursively a descent of N bits, from $L_1 + N$ to L_1 bits of the (head) operands \tilde{x}, \tilde{y} . First we calculate recursively a Euclidean descent of $\lceil N/2 \rceil$ bits to $H = L_1 + \lfloor N/2 \rfloor$ bits in (C4) and (C5).

If $|\tilde{y}| < 2^H$ in (C3) then no descent is calculated with CIDESCENT, but we get a remainder r with $|r| < 2^{H-\frac{1}{2}} < 2^H$ after one Euclidean step. If $|r| \geq 2^{L_1}$, then further iterations of the while-loop are not necessary because of $\max(|\tilde{y}|, |r|) < 2^H$. Otherwise, i.e., if $|r| < 2^{L_1}$, we apply a size modification such that $u' := \tilde{y}$, $v' := r + \varepsilon \tilde{y}$ ($\varepsilon \in \{-1, +1\}$), and $|v'| \geq |\tilde{y}|$. Then $|u' - \varepsilon v'| < 2^{L_1}$, and it follows that no more iteration of the loop is done.

If algorithm CIDESCENT was called recursively in (C4) then there exists $\varepsilon \in \{-1, +1\}$ such that $|u' + \varepsilon v'| < 2^H$. Then the calculation of a Euclidean step leads to a remainder less than 2^H . Either there has to be applied a size modification of the remainder (because of the size bound 2^{L_1}) which causes the termination of the loop, or another Euclidean step is calculated. Now no further iteration is done because either a size modification is done or both operands are less than 2^H such that the maximum condition is not valid anymore.

If the condition $\min(|u' - v'|, |u' + v'|) < 2^{L_1}$ leads to the termination of the while-loop (C5) then the if-condition is fulfilled; hence no second descent CIDESCENT (C6) is calculated. In the case $T = 0$ the condition in (C8) is not satisfied. Moreover, we have $L = L_1$, hence no iteration in (C9) is calculated.

Fig. 2. Euclidean descent in $\mathbb{Z}[i]$: splitting with tails

- (v) Now we consider the case $T > 0$; then $L > N + 5$. In (C2), we have chosen T and L_1 such that $T + L_1 = L + 1$. The steps from (C3) to (C7) calculate a descent from x'', y'' to \tilde{u}, \tilde{v} . There exists an $\varepsilon \in \{-1, +1\}$ such that

$$|\tilde{u}|, |\tilde{v}| \geq 2^{L_1} > |\tilde{u} + \varepsilon \tilde{v}|. \quad (9)$$

After (C8), we have calculated u, v as

$$\begin{aligned} u &:= \tilde{u} \cdot 2^T + \det M \cdot (d \cdot x' - b \cdot y'), \\ v &:= \tilde{v} \cdot 2^T + \det M \cdot (-c \cdot x' + a \cdot y'). \end{aligned} \quad (10)$$

Because of $|x''|, |y''| < 2^{L_1+N-1}$, because of the lower bound of the size of \tilde{u}, \tilde{v} and because of (ii), the coefficients of the cofactor matrix M (descent from x'', y'' to \tilde{u}, \tilde{v}) are bounded by

$$\left(4 + \frac{5}{\sqrt{2}}\right) \cdot \frac{2^{L_1+N-1}}{2^{L_1}} < 2^{N+2} \quad (\text{cf. Theorem 2.1}). \quad (11)$$

Furthermore, there are $|x'|, |y'| \leq 2^{T+\frac{1}{2}}$, that we can estimate u, v using (10) by

$$|u|, |v| > 2^{L_1+T} - 2 \cdot 2^{N+2} \cdot 2^{T+\frac{1}{2}} = 2^{L_1+T} - 2^{L_1+T-1.5} = 2^L (2 - 2^{-0.5}) > 2^L$$

because of $L_1 = N + 5$ and $T + L_1 = L + 1$. Therefore $|u|, |v|$ can never become too small. On the other hand, it follows from (10) — using the size bound (9) — that

$$\begin{aligned} |u + \varepsilon v| &= |(\tilde{u} + \varepsilon \tilde{v}) \cdot 2^T + \det M \cdot (d - \varepsilon c)x' + \det M \cdot (-b + \varepsilon a)y'| \\ &< 2^{L_1+T} + 4 \cdot 2^{N+2} \cdot 2^{T+\frac{1}{2}} \\ &\leq 2^{L+1} + 2^{N+T+4.5} = 2^{L+1} (1 + 2^{-\frac{1}{2}}) < 3.5 \cdot 2^L. \end{aligned}$$

Therefore the number of Euclidean steps in (C9) is bounded by 5, because one of the operands is smaller than $3.5 \cdot 2^L$ in absolute value after one Euclidean step, and after less than four further Euclidean steps we achieve a remainder less than $(2^{-\frac{1}{2}})^4 \cdot 3.5 \cdot 2^L < 2^L$, such that a size modification has to be done. Then no further iteration of the while-loop is executed and the algorithm terminates. \square

Definition 3.1. $\mu_{\mathbb{Z}[i]}(s)$ denotes a smooth upper time bound for the multiplication time of two Gaussian integers bounded by 2^s . Using an asymptotically fast multiplication (cf. [SS71]), we achieve $\mu_{\mathbb{Z}[i]}(s) \leq O(\mu(s)) \leq O(s \log s \log \log s)$.

Theorem 3.2. Let $t(l, n)$ denote the maximum running time of algorithm CIDESCENT(x, y, L) for any $L \leq l$ and arbitrary Gaussian integers x, y with $|x|, |y| < 2^{L+N}$, where $N \leq n - 5$. Then

$$t(l, n) \leq O(\mu_{\mathbb{Z}[i]}(l + n) \cdot \log(n + 1)).$$

Proof. Except for the recursive calls (C4) and (C6), our algorithm requires only a bounded number of arithmetic operations with Gaussian integers of less than 2^{l+n} in absolute value. Since the case $L > N + 5$ is reduced to a problem with parameters $L_1 = N + 5$, $N_1 = N - 1$, we have the general estimate

$$t(l, n) \leq t(n, n) + O(\mu_{\mathbb{Z}[i]}(l + n)). \quad (12)$$

Moreover, the *divide-and-conquer* structure of the algorithm yields the recursive estimate

$$t(2n, 2n) \leq 2t(n, n) + O(\mu_{\mathbb{Z}[i]}(4n)). \quad (13)$$

(Overall one calculates a descent of $2n$ bits with respect to the operand's absolute value by calculating recursively up to two descent of n bits each. At such a recursive call one splits the operands and uses only the heads; hence the running time is bounded by $t(n, n)$.) The estimate (13), together with (12), implies the overall bound

$$t(l, n) \leq O(\mu_{\mathbb{Z}[i]}(l + n) \cdot \log(n + 1)).$$

□

This running time for parameter $l = 0$ is an upper bound for the running time of the GCD calculation of Gaussian integers x, y with $|x|, |y| < 2^n$ because CIDESCENT($x, y, 0$) calculates the GCD (unique up to units) of x, y (see Proposition 2.2).

We can calculate the GCD of integers \mathbb{Z} using a GCD calculation for $\mathbb{Z}[i]$. (By doing this we will put too much effort in our calculation, but for our purpose to show an upper bound of the running time it will be sufficient.) A least remainder Euclidean step in $\mathbb{Z}[i]$ of operands $x, y \in \mathbb{Z}$ yields a remainder $x - qy \in \mathbb{Z}$, thus we can calculate the GCD in \mathbb{Z} (unique up to the sign) by calculating a Euclidean descent in $\mathbb{Z}[i]$.

Corollary 3.1. GCD computation of integers bounded by 2^n in absolute value can be done in time $O(\mu(n) \log n)$. □

Furthermore, a GCD calculation in \mathbb{Z} can be done by a constant factor faster using some special features of \mathbb{Z} that $\mathbb{Z}[i]$ does not have (see [Sch87, SGV94]).

4 Fast Computation of the Biquadratic Residue Symbol

An application of fast GCD calculation in \mathbb{Z} is the fast calculation of the Jacobi symbol. Meyer and Sorenson [MS98] use the right-shift and the left-shift k -ary GCD algorithm to compute the Jacobi symbol of integers bounded by 2^n in absolute value in time $O(n^2 / \log n)$. However, some years ago, Schönhage [Sch87, SGV94] has shown that the Jacobi symbol can be computed in time $O(\mu(n) \log n)$ using the asymptotically fast GCD algorithm for \mathbb{Z} (see [SGV94, p. 245]). Based on such an idea, we will give a brief overview how to compute the biquadratic residue symbol asymptotically fast using the GCD algorithm in $\mathbb{Z}[i]$ presented above. Weilert [Wei99b] has explained that in detail. For number theoretic background, we refer to Ireland and Rosen [IR90], to Lemmermeyer [Lem99] for reciprocity laws and to Neukirch [Neu99, Chapters V.3, VI.8] for Hilbert symbols.

Definition 4.1. We call $x \in \mathbb{Z}[i]$ primary iff $x \equiv 1 \pmod{2+2i}$.

If x is primary then $(1+i) \nmid x$. For a Gaussian integer x that is not divisible by $1+i$, there exists a unique $\varepsilon \in \mathbb{Z}[i]^\times$ such that $\varepsilon \cdot x$ is primary.

Definition 4.2. Let $\pi \in \mathbb{Z}[i]$ be prime, $N(\pi) \neq 2$, where N denotes the norm of the field extension $\mathbb{Q}(i)/\mathbb{Q}$. The biquadratic residue character is defined as

$$\left[\frac{x}{\pi} \right] = \begin{cases} i^j & \text{if } \pi \nmid x \text{ and } j \text{ is the unique } j \in \{0, 1, 2, 3\} \\ & \text{such that } x^{\frac{N(\pi)-1}{4}} \equiv i^j \pmod{\pi}, \\ 0 & \text{if } \pi \mid x. \end{cases}$$

Definition 4.3. Let $x \in \mathbb{Z}[i] \setminus (\mathbb{Z}[i]^\times \cup \{0\})$, $x \nmid (1+i)$. Let $y \in \mathbb{Z}[i]$. There exists a unique prime decomposition of x (up to units), $x = \prod_j \pi_j$, in finitely many prime factors π_j . Define the biquadratic residue symbol as

$$\left[\frac{y}{x} \right] = \begin{cases} \prod_j \left[\frac{y}{\pi_j} \right] & \text{if } x \text{ and } y \text{ are coprime,} \\ 0 & \text{otherwise.} \end{cases}$$

Using some facts about $\left[\frac{\cdot}{\pi_j} \right]$ it can easily be shown that the biquadratic residue symbol is well-defined.

Theorem 4.1. Let $x = x_0 + ix_1$, $y = y_0 + iy_1$ be primary Gaussian integers, x, y coprime. Then we get the biquadratic reciprocity law

$$\left[\frac{x}{y} \right] \left[\frac{y}{x} \right]^{-1} = (-1)^{\frac{N(x)-1}{4} \cdot \frac{N(y)-1}{4}} = (-1)^{\frac{x_0-1}{2} \cdot \frac{y_0-1}{2}} = (-1)^{\frac{x_1 y_1}{4}}.$$

Furthermore, we have the supplementary laws

$$\left[\frac{i}{x} \right] = i^{\frac{1-x_0}{2}}, \quad \left[\frac{1+i}{x} \right] = i^{\frac{x_0-x_1-x_1^2-1}{4}}, \quad \left[\frac{2}{x} \right] = i^{\frac{-x_1}{2}}.$$

Proof. [Lem99, § 6] □

For convenience, we set $\lambda := 1+i$. From Theorem 4.1 it follows that the value of the biquadratic residue symbol depends only on the coset of $\mathbb{Z}[i]/\lambda^f\mathbb{Z}[i]$ of the operands x, y with $f = 7$ ($\lambda^7 = 8 - 8i$).

Now we consider the number field $\mathbb{Q}(i)$ as the smallest cyclotomic field that contains the fourth roots of unity. Every element $a \in \mathbb{Q}(i)$ can be written as $a = \lambda^{v_\lambda(a)} a^*$ with a^* coprime to λ , where v_λ denotes the valuation with respect to the prime ideal $\lambda\mathbb{Z}[i]$. If we denote the Hilbert symbol of a and b in $\mathbb{Q}(i)$ at the prime λ with $(a, b)_\lambda$ we get a generalization of the biquadratic reciprocity law¹ as

$$\left[\frac{y}{x} \right] = (x, y)_\lambda \left[\frac{x}{y^*} \right]. \quad (14)$$

Consider a division chain $x = qy + z$, $y = \tilde{q}z + u$, where x is not divisible by λ . If y is not divisible by λ either, then $y = y^*$, and (14) implies

$$\left[\frac{y}{x} \right] = (x, y)_\lambda \left[\frac{x}{y^*} \right] = (x, y)_\lambda \left[\frac{z}{y} \right].$$

Otherwise, if y is divisible by λ , then z is not divisible by λ , and we get

$$\begin{aligned} \left[\frac{y}{x} \right] &= (x, y)_\lambda \left[\frac{x}{y^*} \right] = (x, y)_\lambda \left[\frac{z}{y^*} \right] \\ &= (x, y)_\lambda (z, y)_\lambda^{-1} \cdot \left[\frac{y}{z} \right] = (x/z, y)_\lambda \cdot \left[\frac{u}{z} \right]. \end{aligned}$$

Because the Hilbert symbol depends only on the cosets $\mathbb{Z}[i]/\lambda^f\mathbb{Z}[i]$ of the operands, it is sufficient to know the operands x, z modulo λ^f , and y modulo λ^{2f-1} . In the case of $y \equiv 0 \pmod{\lambda^f}$ we see that $x/z \equiv 1 \pmod{\lambda^f}$, hence $(x/z, y)_\lambda = 1$. Otherwise, $y = \lambda^k y^*$ with $1 \leq k < f$. We know k exactly and y^* modulo λ^f since y is known modulo λ^{2f-1} . Using

$$(x/z, y)_\lambda = (x/z, \lambda^k)_\lambda \cdot (x/z, y^*)_\lambda,$$

we can calculate the Hilbert symbol $(x, y)_\lambda$. A more sophisticated analysis, presented in [Wei99b, Lemma 2.13, Lemma 2.14] shows that it is already sufficient to know the operands x, y, q modulo $8 \sim \lambda^6$, but this result exceeds the topic of this chapter.

With this we have proved the following theorem which represents the theoretical background for the asymptotically fast computation of the biquadratic residue symbol.

¹ See the general reciprocity law of the n -th power residue symbols, e.g. [Neu99, Theorem VI.8.3], and use the fact that λ is the only prime that divides 4. Every embedding of $\mathbb{Q}(i)$ is totally complex such that the infinite primes need not to be considered.

Theorem 4.2. Let $x_{j-1}, x_j \in \mathbb{Z}[i]$ be coprime, $v_\lambda(x_{j-1}) = 0$. Let $x_{j-1} = q_j x_j + x_{j+1}$, $x_j = q_{j+1} x_{j+1} + x_{j+2}$ be two steps of a division chain (or, respectively, of a Euclidean descent). Then

$$\left[\frac{x_j}{x_{j-1}} \right] = \begin{cases} \left[\frac{x_{j+1}}{x_j} \right] \cdot (x_{j-1}, x_j)_\lambda & \text{if } \lambda \nmid x_j, \\ \left[\frac{x_{j+2}}{x_{j+1}} \right] \cdot (x_{j-1}/x_{j+1}, x_j)_\lambda & \text{if } \lambda \mid x_j, \end{cases}$$

and the Hilbert symbols $(x_{j-1}, x_j)_\lambda$ resp. $(x_{j-1}/x_{j+1}, x_j)_\lambda$ used above depend only on the $\mathbb{Z}[i]/\lambda^{13}\mathbb{Z}[i]$ -cosets of the operands x_{j-1}, x_j, q_j . \square

Finally, we claim that such a reduction from $\left[\frac{x_j}{x_{j-1}} \right]$ to another biquadratic residue symbol as in Theorem 4.2 always leads to a symbol with “denominator” coprime to λ . Therefore we can use the reduction of Theorem 4.2 again and again for the whole Euclidean descent.

Lemma 4.1. Let $x, y \in \mathbb{Z}[i]$, $\gcd(x, y) \sim 1$. Set $x_0 := x$, $x_1 := y$. Let $x_{j-1} = q_j x_j + x_{j+1}$ be a Euclidean step with $1 \leq j \leq r$ and $x_r \sim x_{r+1} \sim 1$ (i.e., the last Euclidean step of the GCD calculation is modified such that the remainder is a unit). Then

$$(1+i) \mid x_j \implies (1+i) \nmid x_{j-1}, x_{j+1} \quad (1 \leq j \leq r).$$

Proof. [Wei99b, Lemma 2.16] \square

Now we have laid down the basics on which we can easily derive the fast algorithm for the computation of the biquadratic residue symbol from the previous considerations. For a Gaussian integer $z \in \mathbb{Z}[i]$, we denote the coset of z in $\mathbb{Z}[i]/\lambda^f\mathbb{Z}[i]$ with z' .

Theorem 4.3. Let $x, y \in \mathbb{Z}[i]$, x not divisible by λ . Let the real and imaginary parts of x and y be bounded by 2^n in absolute value. Then the running time of algorithm QUARTIC is bounded by $O(\mu(n) \log n)$.

In particular, the steps from (Q3) to (Q8) of the algorithm require only running time $O\left(\sum_j \text{size}(q_j)\right)$ where $\text{size}(q_j) \approx O(\log |q_j|)$ denotes the number of bits for a binary coding of the real and imaginary part of q_j in a standard manner.

Proof. The calculation of the GCD in step (Q2) can be achieved in running time $O(\mu(n) \log n)$.

The extraction of q'_j from the coding of q_j can be done as fast as q_j can be read, i.e., in time $O(\text{size}(q_j))$. Then the calculation of x'_{j-1} (Q5) requires only constant running time because the operands q'_j, x'_j, x'_{j+1} are bounded by $2 \cdot |\lambda|^{13}$. The test whether λ is a divisor of x'_{j-1} , can be accomplished by comparing the lowest bits of the real and imaginary part of x'_{j-1} , hence can be done in constant time. The calculation of the Hilbert symbols is possible in constant time (e.g., table lookup) because the operands (as representatives of the coset $\mathbb{Z}[i]/\lambda^{13}\mathbb{Z}[i]$) are bounded. This proves the claimed overall estimate for the running time. \square

Algorithm QUARTIC(x, y)

For Gaussian integers $x, y \in \mathbb{Z}[i]$, x not divisible by λ , the algorithm QUARTIC calculates the biquadratic residue symbol $\left[\frac{y}{x} \right]$.

Q1: $x_0 := x, x_1 := y;$
 Q2: Calculate a Euclidean descent (using the fast GCD algorithm CIDESCENT with parameter $L = 0$) from x_0, x_1 , consisting of Euclidean steps $x_{j-1} = q_j x_j + x_{j+1}$, $1 \leq j \leq r$, to x_r , where the condition to stop is $x_r \sim x_{r+1}$; thereby store the q_j 's for later use.
 Q3: **if** $x_r \not\sim 1$ **then**
 return 0;
 else $\setminus\setminus \gcd(x, y) \not\sim 1$
 else $\setminus\setminus \gcd(x, y) \sim 1$
 Q4: $j := r, s := \left[\frac{x'_{r+1}}{x'_r} \right] = 1;$
 Q5: **while** $j > 0$ **do** $\setminus\setminus$ invariant $s = \left[\frac{x'_{j+1}}{x'_j} \right]$
 $x'_{j-1} := q'_j x'_j + x'_{j+1};$
 if $(1+i) \mid x'_{j-1}$ **then** $\setminus\setminus j \geq 2$ because of $(1+i) \nmid x$
 Q6: $x'_{j-2} := q'_{j-1} x'_{j-1} + x'_j;$
 $s := s \cdot (x'_{j-2}/x'_j, x'_{j-1})_\lambda, j := j - 2;$ $\setminus\setminus j \geq 0$
 else
 Q7: $s := s \cdot (x'_{j-1}, x'_j)_\lambda, j := j - 1;$ $\setminus\setminus j \geq 0$
 Q8: **return** $s;$

5 Practical Running Time

In this section we will discuss the running time of an implementation of our new GCD algorithm, in comparison with three other GCD algorithms in $\mathbb{Z}[i]$ (standard Euclidean algorithm, Lehmer-type algorithm and $(1+i)$ -adic GCD algorithm, i. e., a GCD algorithm in $\mathbb{Z}[i]$ as an analogue to the binary GCD algorithm in \mathbb{Z} ; cf. [Wei00]).

5.1 The Machine Model TP

Our machine model is the multitape Turing machine TP, described in detail in Schönhage et al. [SGV94]. This machine model is not only a theoretical model, but it can be realized on real computers, i. e., the Turing programs can be executed on existing hardware. Such an execution on a real computer allows the use of implemented fast TP algorithms for calculations.

The alphabet Σ of the Turing machine TP consists of 32-bit binary words (today's workstations or personal computers have at least a 32-bit CPU). Mostly such a binary word is interpreted as a *digit* of an integer with base 2^{32} . To avoid large Turing tables, the low-level instructions in each Turing step are often arithmetical operations with 32-bit-words.

The programs for the Turing machine TP are written in an assembly language TPAL as a representation for the Turing table. There are programs for

operations with long integers, coded by word sequences (for a detailed description see [SGV94]). In particular, a fast multiplication SML is implemented in TPAL with running time of $\mu(n) \leq O(n \log n \log \log n)$ for n word integers.

The machine model allows a quite exact running time analysis, which matches well with the running time on today's computers. Each elementary operation (i. e., instructions for one Turing step) corresponds to a hypothetical time value, measured in *time units* (tu), and the running time of an algorithm is the sum of the time units of the instructions executed. The elementary load, store and calculation operations and (conditional) jumps are valued as 1 tu or 2 tu; 32×32 bit multiplication instructions are valued as 33 tu, division instructions as 66 tu. Using a TP implementation on an Intel Pentium II/400 MHz computer under the Linux operating system, we achieve a performance of about 300 Mtu/sec, where 1 Mtu ("Mega time unit") stands for one million time units.

5.2 Comparison of the Running Time of Several GCD Algorithms in $\mathbb{Z}[i]$

In Table 1, we have listed the running times of the different GCD algorithms. For these timing tests, we have generated at random Gaussian integers with real and imaginary parts each consisting of n words, i. e., real and imaginary parts are bounded approximately by 2^{32n} in absolute value. We chose the size n in order to achieve a good overview of the running time's behaviour of the GCD algorithms.

The "ordinary" algorithm is an implementation of the Euclidean algorithm, which calculates the least remainder in every Euclidean step. Thus in every Euclidean step, the algorithm exactly calculates a remainder-division. The number of Euclidean steps is bounded by $O(n)$, so that we achieve a running time of $O(n \cdot \mu(n))$.

The so-called "Lehmer-type" algorithm is asymptotically faster than the ordinary Euclidean algorithm. This algorithm approximately calculates the quotients of the two operands in each Euclidean step. The absolute value of the new calculated operand is at least reduced by a factor of 0.85. Therefore the number of Euclidean steps is bounded by $O(n)$, hence the running time is bounded by $O(n^2)$.

The $(1+i)$ -adic algorithm is an analogue to the binary GCD algorithm in \mathbb{Z} . It reduces the intermediate operands by powers of $1+i$. It can be shown (see [Wei00]) that every iteration yields a reduction of the sum of the operand's norm at least by the factor 0.65. In every step, arithmetical operations with running time $O(n)$ are done, and the number of iterations is bounded by $O(n)$; therefore we get $O(n^2)$ as an upper bound for the running time. It remains to mention that this algorithm is not able (in an obvious manner) to calculate cofactors for a representation of the GCD.

The algorithm CIDESCENT, based on the controlled Euclidean descent in $\mathbb{Z}[i]$, was discussed in detail above. Its running time is bounded by $O(\log n \cdot \mu(n))$ (cf. Theorem 3.2).

Table 1. Running time of several GCD algorithms in $\mathbb{Z}[i]$ (in Mtu[†])

Size n words	algorithm for the calculation of a GCD	ordinary	Lehmer-type	(1 + i)-adic	CIDESCENT
1		0.1	0.1	0.0	0.1
4		0.5	0.4	0.3	0.7
10		3.2	1.2	0.8	2.4
50		164.2	15.5	6.7	17.4
100		933.7	52.7	20.1	41.9
150		2544.6	113.2	40.5	71.1
200		5124.8	201.1	66.5	101.0
250		8337.3	303.8	100.1	138.6
300		12709.4	432.0	142.0	176.8
350		18015.2	590.1	187.9	215.2
400		24236.2	760.1	243.5	250.7
410		25647.9	793.8	253.3	253.3
420		27503.1	846.5	265.5	268.5
430		28793.7	877.3	279.1	275.8
440		30401.5	924.1	290.6	284.7
450		32056.6	968.6	304.6	280.2
500		39474.8	1186.8	369.5	357.4
600		59628.5	1706.1	526.2	426.8
700		83112.2	2295.2	709.2	523.1
800		112156.2	3013.2	916.9	618.6
900		142734.7	3783.3	1153.7	723.6
1000		174286.3	4648.5	1417.0	827.3
2000		837500.9	18615.7	5536.3	2106.7

[†] We achieve a performance of about 300 Mtu/sec on an Intel Pentium II/400 MHz.

Comparing the running times of these algorithms, one observes that the $(1+i)$ -adic algorithm is faster than the ordinary Euclidean and the Lehmer-type algorithm for any size. Moreover, if the real or imaginary part consist of $n \geq 420$ words, the algorithm CIDESCENT is faster than the $(1+i)$ -adic algorithm.

The idea our algorithm is based on can be transferred to the five norm-Euclidean rings of algebraic integers of imaginary quadratic number fields (including $\mathbb{Z}[i]$) in a generalized manner, but this is beyond the scope of this article. In particular, with other methods than in [CC76], one can bound the size of the cofactor matrix for these five rings, too.

References

- [Bre76] R. P. Brent, *Analysis of the binary Euclidean algorithm*, Algorithms and Complexity: New Directions and Recent Results (J. F. Traub, ed.), Academic Press, New York, 1976, pp. 321–355.
- [BS96] E. Bach and J. Shallit, *Algorithmic Number Theory, Volume I: Efficient Algorithms*, Foundations of Computing, MIT Press, Cambridge, MA, USA, 1996.
- [Cav73] B. F. Caviness, *A Lehmer-Type Greatest Common Divisor Algorithm for Gaussian Integers*, SIAM Review **15** (1973), no. 2, 414.
- [CC76] B. F. Caviness and G. E. Collins, *Algorithms for Gaussian Integer Arithmetic*, Proceedings of the ACM Symposium on Symbolic and Algebraic Computation, Yorktown Heights (1976), 36–45.
- [Hea56] T. L. Heath, *The Thirteen Books of Euclid's Elements*, second ed., vol. 2, Cambridge University Press, New York, 1956, Books III–IX.
- [IR90] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, second ed., Graduate Texts in Mathematics, vol. 84, Springer-Verlag, Berlin, Heidelberg, New York, 1990.
- [Jeb93] T. Jebelean, *A Generalization of the Binary GCD Algorithm*, ISSAC'93: Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation (New York) (M. Bronstein, ed.), ACM Press, 1993, pp. 111–116.
- [Knu98] D. E. Knuth, *Seminumerical Algorithms*, third ed., The Art of Computer Programming, vol. 2, Addison-Wesley, Reading, MA, USA, 1998.
- [KR89] E. Kaltofen and H. Rolletschek, *Computing Greatest Common Divisors and Factorizations in Quadratic Number Fields*, Math. Comp. **53** (1989), no. 188, 697–720.
- [Leh38] D. H. Lehmer, *Euclid's Algorithm for Large Numbers*, Amer. Math. Monthly **45** (1938), 227–233.
- [Lem95] F. Lemmermeyer, *The Euclidean Algorithm in Algebraic Number Fields*, update of a version published in Expo. Math. **13** (1995), 385–416, 1995.
- [Lem99] F. Lemmermeyer, *Reciprocity Laws. Their Evolution from Euler to Artin*, unpublished manuscript, <http://www.rzuser.uni-heidelberg.de/~hb3/rec.html>, 1999.
- [Len74] H. W. Lenstra Jr., *Lectures on Euclidean Rings*, Bielefeld, 1974.
- [MS98] S. M. Meyer and J. P. Sorenson, *Efficient Algorithms for Computing the Jacobi Symbol*, J. Symbolic Comput. **26** (1998), no. 4, 509–523.
- [Neu99] J. Neukirch, *Algebraic Number Theory*, Springer-Verlag, Berlin, Heidelberg, New York, 1999.
- [Nor85] G. H. Norton, *Extending the binary gcd algorithm*, AAECC-3 (J. Calmet, ed.), Lecture Notes in Computer Science, vol. 229, 1985, pp. 363–372.
- [Nor89] G. Norton, *A Shift-Remainder GCD Algorithm*, Proceedings of the 5th International Conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes AAECC-5 (Berlin, Heidelberg, New York) (L. Huguet and A. Poli, eds.), Lecture Notes in Computer Science, vol. 356, Springer-Verlag, 1989, pp. 350–356.
- [Rol86] H. Rolletschek, *On the Number of Divisions of the Euclidean Algorithm Applied to Gaussian Integers*, J. Symbolic Comput. **2** (1986), 261–291.
- [Rol89] H. Rolletschek, *Shortest division chains in imaginary quadratic number fields*, Symbolic and algebraic computation: International Symposium ISSAC '88, Rome, Italy, July 4–8, 1988: proceedings (Berlin, Heidelberg, London etc.) (P. (Patrizia) Gianni, ed.), Lecture Notes in Computer Science, vol. 358, Springer Verlag, 1989, Conference held jointly with AAECC-6., pp. 231–243.

- [Rol90] H. Rolletschek, *Shortest Division Chains in Imaginary Quadratic Number Fields*, J. Symbolic Comput. **9** (1990), 321–354.
- [Sch71] A. Schönhage, *Schnelle Berechnung von Kettenbruchentwicklungen*, Acta Informatica **1** (1971), 139–144.
- [Sch87] A. Schönhage, *IGCDOC, Computation of integer gcd's*, unpublished manuscript, 1987.
- [Sch91] A. Schönhage, *Fast Reduction and Composition of Binary Quadratic Forms*, Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC '91 (S. M. Watt, ed.), ACM Press, 1991, pp. 128–133.
- [SGV94] A. Schönhage, A. F. W. Grotfeld, and E. Vetter, *Fast Algorithms – A Multitape Turing Machine Implementation*, BI Wissenschaftsverlag, Mannheim, 1994.
- [SL97] M. S. Sedjelmaci and C. Lavault, *Improvements on the Accelerated Integer GCD Algorithm*, Inform. Process. Lett. **61** (1997), 31–36.
- [Sor90] J. P. Sorenson, *The k-ary GCD Algorithm*, Technical Report CS-TR-90-979, University of Wisconsin, Madison, 1990.
- [Sor94] J. Sorenson, *Two Fast GCD Algorithms*, J. Algorithms **16** (1994), no. 1, 110–144.
- [Sor95] J. Sorenson, *An Analysis of Lehmer's Euclidean GCD Algorithm*, Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation ISSAC'95 (1995), 254–258.
- [SS71] A. Schönhage and V. Strassen, *Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292.
- [SS94a] J. Shallit and J. Sorenson, *Analysis of a Left-Shift Binary GCD Algorithm*, Proceedings of the First International Symposium “Algorithmic Number Theory” ANTS-I (Berlin, Heidelberg, New York) (L. M. Adleman and M.-D. Huang, eds.), Lecture Notes in Computer Science, vol. 877, Springer-Verlag, 1994, pp. 169–183.
- [SS94b] J. Shallit and J. Sorenson, *Analysis of a Left-Shift Binary GCD Algorithm*, J. Symbolic Comput. **17** (1994), no. 6, 473–486.
- [Ste67] J. Stein, *Computational problems associated with Racah algebra*, J. Comput. Phys. **1** (1967), 397–405.
- [Val98] B. Vallée, *The Complete Analysis of the Binary Euclidean Algorithm*, Proceedings of the Third International Symposium “Algorithmic Number Theory” ANTS-III (Berlin, Heidelberg, New York) (J. P. Buhler, ed.), Lecture Notes in Computer Science, vol. 1423, Springer-Verlag, 1998, pp. 77–94.
- [Web95] K. Weber, *The accelerated integer GCD algorithm*, ACM Trans. Math. Software **21** (1995), 111–122.
- [Wei99a] A. Weilert, *Asymptotisch schnelle g.g.T.-Berechnung im Ring der Gaußschen Zahlen $\mathbb{Z}[i]$* , Diplomarbeit, Institut für Informatik, Universität Bonn, 1999.
- [Wei99b] A. Weilert, *Ein schneller Algorithmus zur Berechnung des quartischen Restsymbols*, Diplomarbeit, Mathematisches Institut, Universität Bonn, 1999.
- [Wei00] A. Weilert, *(1 + i)-adic GCD Computation in $\mathbb{Z}[i]$ as an Analogue to the Binary GCD Algorithm*, Proceedings of the Seventh Rhine Workshop on Computer Algebra RWCA'00 (T. Mulders, ed.), 2000, Bregenz, Austria, March 22–24, pp. 1–13.

Author Index

- Arita, Seigo, 113
Auer, Roland, 127
- Backes, Werner, 135
Blackburn, Simon R., 153
Bruin, Nils, 169
- Cai, Jin-Yi, 1
Cantor, David G., 185
Cavallar, Stefania, 209
Cheng, Qi, 233
Chinta, Gautam, 247
Cohen, Henri, 257, 269
- Diaz y Diaz, Francisco, 257, 269
- Elkies, Noam D., 33
- Fieker, Claus, 285
Flynn, E. Victor, 65
Friedrichs, Carsten, 285
- Galway, William F., 297
Gaudry, Pierrick, 313
Geemen, Bert van, 333
Gordon, Daniel M., 185
Gunnells, Paul E., 247, 347
- Harasawa, Ryuichi, 359
Harley, Robert, 313
Huang, Ming-Deh A., 233, 377
- Joux, Antoine, 385
- Kohel, David R., 395, 405
Kueh, Ka Lam, 377
- Louboutin, Stéphane, 413
Müller, Siguna, 423
- Nagao, Koh-ichi, 439
Nguyen, Phong Q., 85
- Olivier, Michel, 257, 269
Omar, Sami, 449
- Pinch, Richard G.E., 459
- Richstein, Jörg, 475
Roblot, Xavier-François, 491
Rodriguez-Villegas, Fernando, 505
- Scheidler, Renate, 515
Sczech, Robert, 247
Shparlinski, Igor E., 395
Smith, Derek A., 533
Sorenson, Jonathan P., 539
Stein, William A., 405
Stern, Jacques, 85
Stroeker, Roel J., 551
Suzuki, Joe, 359
- Tan, Ki-Seng, 377
Tangedal, Brett A., 491
Teske, Edlyn, 153, 563
Top, Jaap, 333
Tzanakis, Nikolaos, 551
- Vollmer, Ulrich, 581
- Weilert, André, 595
Wetzel, Susanne, 135
Williams, Hugh C., 563