Open access • Dissertation • DOI:10.14264/UQL.2014.506

On the number field sieve: polynomial selection and smooth elements in number fields — Source link

Nicholas Vincent Coxon

Published on: 01 Jun 2012

Topics: General number field sieve, Irreducible polynomial, Algebraic number, Polynomial and Reciprocal polynomial

THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

On the Number Field Sieve:

Polynomial Selection and Smooth Elements in Number Fields

Nicholas Vincent Coxon

BSc (hons)

A thesis submitted for the degree of Doctor of Philosophy at The University of Queensland in June 2012

School of Mathematics and Physics

Abstract

The number field sieve is the asymptotically fastest known algorithm for factoring large integers that are free of small prime factors. Two aspects of the algorithm are considered in this thesis: polynomial selection and smooth elements in number fields. The contributions to polynomial selection are twofold. First, existing methods of polynomial generation, namely those based on Montgomery's method, are extended and tools developed to aid in their analysis. Second, a new approach to polynomial generation is developed and realised. The development of the approach is driven by results obtained on the divisibility properties of univariate resultants.

Examples from the literature point toward the utility of applying decoding algorithms for algebraic error-correcting codes to problems of finding elements in a ring with a smooth representation. In this thesis, the problem of finding algebraic integers in a number field with smooth norm is reformulated as a decoding problem for a family of error-correcting codes called NF-codes. An algorithm for solving the weighted list decoding problem for NF-codes is provided. The algorithm is then used to find algebraic integers with norm containing a large smooth factor. Bounds on the existence of such numbers are derived using algorithmic and combinatorial methods.

Declaration by the Author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications During Candidature

[47] Nicholas Coxon. On nonlinear polynomial selection for the number field sieve. ArXiv e-Print archive, `arXiv:1109.6398 [math.NT]`, September 2010. http://arxiv.org/abs/1109.6398.

[48] Nicholas Coxon. List decoding of number field codes. *Designs, codes and cryptography*, 2013. doi:10.1007/s10623-013-9803-x.

Publications Included in the Thesis

Publications [47] and [48] have been incorporated into Chapter 3 and Chapter 5, respectively.

Contributions by Others to the Thesis

As supervisor, Victor Scharaschkin helped guide the research presented in the thesis. There has been no contribution by others to the writing of the thesis.

Statement of Parts of the Thesis Submitted to Qualify for the Award of Another Degree

None.

Acknowledgements

I am greatly indebted to Victor Scharaschkin for his supervision during my candidature. The support he has extended to me, and his informed guidance and advice, have been critically important throughout my postgraduate and undergraduate studies. I would like to thank Gary Carter, Eric Mortenson and Graham Norton for helpful discussions and their comments on draft chapters of this thesis.

I would like to acknowledge financial support of the Australian Postgraduate Award.

I have had the pleasure of working with and learning from many of the staff and students of the Mathematics Department. I am proud to call many of you my friends, and wish you all the best in your future endeavours.

It would not have been possible to write this thesis without the help and encouragement of my fellow postgraduate students. In particular, I would like to thank Tristan Dunning, Peter Finch, Samuel Hambleton, Dejan Jovanovic, Geoff Martin, Thomas McCourt and Sian Stafford with whom daily conversations have been constructive, disruptive when needed most, and always greatly appreciated.

To all the friends with whom I have shared many adventures whilst working on this thesis, thank you.

Finally, to my family, thank you for always being there with love and support.

Keywords

Integer factorisation, number field sieve, polynomial selection, smooth numbers, list decoding, number field codes, geometry of numbers, resultants.

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 010101, Algebra and Number Theory, 89.939%
ANZSRC code: 080201, Analysis of Algorithms and Complexity, 10.061%

Fields of Research (FoR) Classification

FoR code: 0101, Pure Mathematics, 100.000%

Contents

List of Tables

Nomenclature

The following is a list of abbreviations and symbols that appear in the thesis. Where possible, a page number is provided to indicate where an abbreviation or symbol first appears. The list of symbols is incomplete, with only those symbols that are not defined elsewhere in the thesis, or that appear in multiple chapters, being listed. All remaining symbols are possibly used for several different purposes throughout the thesis, but are always used consistently within the confines of a single chapter, where the relevant definition is found.

List of Abbreviations

CFRAC	Continued fractions method, p. 1.
GP	Geometric progression, p. 41.
NFS	Number field sieve, p. 1.
QS	Quadratic Sieve, p. 1.
SNFS	Special number field sieve, p. 1.

List of Symbols

$\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$	The set of integers, rationals, real numbers and complex numbers, respectively.
$\mathbb{F}_q$	The finite field of order q, where q is a prime power.
$\mathbb{Z}/n\mathbb{Z}$	The ring of integers modulo n.
$A^\times$	The group of units of a ring A.
$\mathrm{GL}_n(A)$	The general linear group of $n \times n$ invertible matrices with entries in a ring A.
$\langle a_1, \ldots, a_n \rangle$	The ideal generated by elements $a_1, \ldots, a_n$ of a ring.

$\lfloor x \rfloor$, $\lceil x \rceil$, $\lfloor x \rceil$	The floor, ceiling and nearest integer (defined to be $\lfloor x + 1/2 \rfloor$) of $x \in \mathbb{R}$, respectively.
$\mathrm{Re}(x)$, $\mathrm{Im}(x)$	The real and imaginary parts of $x \in \mathbb{C}$, respectively.
$\nu_p(x)$	The p-adic valuation of $x \in \mathbb{Q}$, p. 14.
$\psi(x, y)$	The number of y-smooth integers in the interval $[1, x]$, p. 3.

$\mathrm{lc}(f)$	The leading coefficient of a polynomial f.
$\mathrm{disc}(f)$	The discriminant of a polynomial f.
$\Vert f \Vert_{p,s}$	The skewed p-norm of a polynomial $f \in \mathbb{R}[x]$, p. 18.
$\Vert f \Vert_p$	The skewed p-norm of a polynomial $f \in \mathbb{R}[x]$ in the special case where $s = 1$, p. 19.
$\Vert f \Vert_{L^2,s}$	The skewed L^2-norm of a polynomial $f \in \mathbb{R}[x]$, p. 19.
$\mathrm{Syl}(f_1, f_2)$	The Sylvester matrix of $f_1, f_2 \in \mathbb{A}[x]$, where $\mathbb{A}$ is an integral domain, p. 70.
$\mathrm{Res}(f_1, f_2)$	The resultant of $f_1, f_2 \in \mathbb{A}[x]$, where $\mathbb{A}$ is an integral domain, p. 71.
$\alpha(f, y)$	The α-value over primes up to $y > 0$ of an irreducible polynomial $f \in \mathbb{Z}[x]$, p. 23.

$\mathcal{O}_K$	The ring of algebraic integers in a number field K, p. 9.
D_K	The discriminant of a number field K, p. 113.
$N_K(x)$	The (field) norm of $x \in K$, where K is a number field, p. 9.
$\mathfrak{N}\mathfrak{a}$	The norm of an ideal $\mathfrak{a}$ in the ring of integers of a number field, p. 113.
$\mathrm{size}_{\boldsymbol{s}}(x)$	The $\boldsymbol{s}$-shifted size of $x \in K$, where K is a number field, p. 113.
T_2	The map $T_2 : K \to \mathbb{R}$ defined by $x \mapsto \sum_\sigma \sigma(x)\overline{\sigma(x)}$, where K is a number field and σ ranges over the field embeddings of K in the field $\mathbb{C}$, p. 117.
$\delta_\mathbb{R}$	The Minkowski map, p. 118.

$\langle\ ,\ \rangle$	The usual inner product in $\mathbb{R}^n$, p. 46.
$\Vert . \Vert_2$	The norm on $\mathbb{R}^n$ induced by $\langle\ ,\ \rangle$, p. 40.
$\Vert . \Vert_{2,s}$	For $s > 0$ and $\boldsymbol{x} \in \mathbb{R}^{n+1}$, $\Vert \boldsymbol{x} \Vert_{2,s} = \Vert \boldsymbol{x} S \Vert_2$, where $S = s^{-\frac{n}{2}} \cdot \mathrm{diag}(1, s, \ldots, s^n)$, p. 53.

$\det \Lambda$	The determinant of a lattice $\Lambda \subset \mathbb{R}^n$, p. 40.
$\lambda_i(\Lambda)$	The ith minimum of a lattice $\Lambda \subset \mathbb{R}^n$, p. 46.
γ_n	Hermite's constant for dimension n, p. 47.
E_Λ	The $\mathbb{Q}$-vector subspace of $\mathbb{Q}^n$ generated by any basis of a lattice $\Lambda \subseteq \mathbb{Z}^n$, p. 48.
$\Lambda^\perp$	The orthogonal lattice of a lattice $\Lambda \subseteq \mathbb{Z}^n$, p. 48.
Λ_S	The lattice $\{\boldsymbol{x} \cdot S \mid \boldsymbol{x} \in \Lambda\}$, where $\Lambda \subset \mathbb{R}^n$ is a lattice and $S \in \mathrm{GL}_n(\mathbb{R})$, p. 49.
$(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)_S$	The basis $(\boldsymbol{b}_1 S, \ldots, \boldsymbol{b}_k S)$ for the lattice Λ_S, where $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ is a basis of a lattice $\Lambda \subset \mathbb{R}^n$ and $S \in \mathrm{GL}_n(\mathbb{R})$, p. 49.
$[c_0, \ldots, c_{l-1}]$	A vector with entries that form a geometric progression modulo N, p. 41.
$\mathrm{diag}(a_1, \ldots, a_n)$	The $n \times n$ diagonal matrix with entries $a_1, \ldots, a_n$ on the main diagonal.

The problem of factoring integers is a good one to test our mettle as mathematicians. First it is a fundamental as a problem can be. Second, while having the patina of centuries of history, the problem has taken on a new urgency for its connection with public-key cryptography. Third, it it is a very *hard* problem, but not so hard that we do not occasionally gain an insight and make an advance.

Carl Pomerance [145]

Chapter 1

Introduction

This chapter introduces the number field sieve (NFS) and the two problems that are the focus of this thesis: the selection of polynomials for the number field sieve and finding smooth elements in number fields. The congruence of squares approach to integer factorisation is described. Then concepts central to the number field sieve are introduced through an examination of the Morrison-Brillhart approach to constructing a congruence of squares. This is followed by a brief description of the number field sieve, including a discussion of its asymptotic advantage over previous algorithms. Throughout, special attention is given to the polynomial selection problem and the problem of finding smooth elements in number fields. The chapter concludes with an outline of the remainder of the thesis.

Throughout the thesis, N denotes a positive odd integer that requires factorisation.

1.1 Congruence of Squares Factoring

Congruence of squares factoring algorithms attempt to factor a composite integer N by finding solutions to the congruence $x^2 \equiv y^2 \pmod{N}$. For each solution to the congruence, a factor of N can potentially be obtained by computing $\gcd(x \pm y, N)$. For N containing at least two distinct odd prime factors, the computation of $\gcd(x - y, N)$ yields a factor of N for at least half of the pairs $(x, y) \in \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ with $x^2 \equiv y^2 \pmod{N}$ and $\gcd(xy, N) = 1$. This approach to factoring has been adopted in many algorithms including, but not limited to, Dixon's random squares method [52], Morrison and Brillhart's continued fractions method (CFRAC) [123], Pomerance's quadratic sieve (QS) [143], the special number field sieve (SNFS) [104] and finally the number field sieve (NFS) [29].

At a high level, existing congruence of squares factoring algorithms may each be described by a choice of ring A and homomorphism $\varphi : A \to \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$, together with a method for generating elements in A whose image under φ lies on the *diagonal* $D = \{(x, x) \mid x \in (\mathbb{Z}/N\mathbb{Z})^\times\}$. Such elements, called *relations*, are usually constructed in a random or pseudo-random manner. Then common to existing algorithms is an approach whereby relations are combined multiplicatively to obtain squares in A. For a square $a^2 \in A$, such that $a \in A$ and $\varphi(a^2) \in D$, the image of a yields a congruence of squares: if $\varphi(a) = (x, y)$, then $\varphi(a^2) = (x^2, y^2) \in D$, thus $x^2 \equiv y^2 \pmod{N}$. Therefore, provided such a square $a^2 \in A$, with $\varphi(a) = (x, y)$, computing $\gcd(x \pm y, N)$ may yield a factorisation of N.

The random squares, continued fractions and the quadratic sieve algorithms share a common choice of A and φ: in each algorithm, $A = \mathbb{Z} \times \mathbb{Z}$ and the homomorphism φ is defined by $(u, v) \mapsto (u \bmod N, v \bmod N)$. Further to sharing a common choice of A and φ, the algorithms each employ the Morrison–Brillhart approach [123] to constructing products of relations which form squares in $\mathbb{Z} \times \mathbb{Z}$. In the remainder of this section, the Morrison–Brillhart approach is discussed and a primitive analysis of algorithms based on the approach provided.

1.1.1 The Morrison–Brillhart Approach

The Morrison–Brillhart approach to constructing squares in $\mathbb{Z} \times \mathbb{Z}$ involves combining relations of the form $(u^2, v(u))$, where $v(u)$ denotes the least absolute remainder of u^2 modulo N. This approach appears in algorithms that predate Morrison and Brillhart's continued fractions algorithm. However, it was Morrison and Brillhart who introduced an efficient linear algebra based approach for identifying combinations of relations $(u^2, v(u))$ forming squares in $\mathbb{Z} \times \mathbb{Z}$. The relations $(u^2, v(u))$ combined in their approach are restricted to those where $v(u)$ is a y-smooth integer.

Definition 1.1.1. A nonzero integer v is called *y-smooth* if it has no prime factor exceeding y.

Let $\pi(y)$ denote the number of primes less than or equal to y. To each y-smooth integer v, one can assign a $(\pi(y) + 1)$-dimensional vector $\boldsymbol{v}(v)$, with entries indexed by -1 and the primes $p \leq y$, such that an entry indexed by a prime p contains its exponent in the factorisation of v, and the entry index by -1 contains 0 or 1 depending on whether v is positive or negative respectively. Then a product of nonzero y-smooth integers $v_1, \ldots, v_n$ is a square in $\mathbb{Z}$ if and only if $\sum_{i=1}^{n} \boldsymbol{v}(v_i)$ contains all even entries. Since only the parity of the entries in the final vector is of concern, the exponent vectors can simply be viewed as elements of the vector space $\mathbb{F}_2^{\pi(y)+1}$. Therefore, given nonzero y-smooth integers $v_1, \ldots, v_n$ such that there exists a subset of the v_i's whose product is a square in $\mathbb{Z}$, such a subset may be identified by finding a linear dependence among the exponent vectors $\boldsymbol{v}(v_i) \in \mathbb{F}_2^{\pi(y)+1}$.

For positive integer parameters y and k, the Morrison–Brillhart approach begins with a search to find $\pi(y) + 1 + k$ distinct relations $(u_i^2, v(u_i))$ such that $v(u_i)$ is y-smooth. If the search is successful, then there exists at least k distinct subsets $\mathcal{S}$ of indices such that $\sum_{i \in \mathcal{S}} \boldsymbol{v}(v(u_i)) = \mathbf{0}$ in $\mathbb{F}_2^{\pi(y)+1}$. By performing linear algebra over $\mathbb{F}_2$, $t \geq k$ such subsets $\mathcal{S}_1, \ldots, \mathcal{S}_t$ are identified. For each subset $\mathcal{S}_j$, setting

$$x_j = \prod_{i \in \mathcal{S}_j} u_i \quad \text{and} \quad y_j = \sqrt{\prod_{i \in \mathcal{S}_j} v(u_i)}$$

leads to a congruence of squares $x_j^2 \equiv y_j^2 \pmod{N}$. Consequently, the final step of the approach is to attempt to factor N by computing $\gcd(x_j \pm y_j, N)$, for $1 \leq j \leq t$. The task of finding dependencies among the exponent vectors may be performed by algorithms such as Gaussian elimination (see [93]), structured Gaussian elimination [136, 98, 147, 17], the block Lanczos algorithm [41, 121, 138] or the block Wiedemann algorithm [42, 163]. For a matrix of dimension n, the running time of Gaussian elimination is $O(n^3)$; the running time of the block Lanczos and Wiedemann algorithms is $O(nw)$, where w is the number of nonzero entries in the matrix (called the *weight* of the matrix).

Algorithms based on the Morrison–Brillhart approach differ in how the relations $(u_i^2, v(u_i))$, with $v(u_i)$ y-smooth, are constructed. Existing algorithms such as the random squares method, CFRAC and QS find relations by generating a stream of quadratic residues $v_1, v_2, \ldots$, which are then tested for smoothness. The time taken by this approach then depends on the smoothness probabilities of the residues v_i. For integers for $x, y \geq 1$, define

$$\psi(x,y) = |\{v \in [1,x] \cap \mathbb{Z} \mid v \text{ is } y\text{-smooth}\}|.$$

Canfield et al. [30] showed that

$$\psi\left(x, x^{1/u}\right) = xu^{-u+o(1)},$$

uniformly as $u \to \infty$ and $u < (1-\varepsilon)\log x/\log\log x$. By setting $u = \log x/\log y$, the approximation $\psi(x,y) \approx xu^{-u}$ is obtained for $y > \log^{1+\varepsilon} x$ and x large. Suppose the residues v_i all lie in some interval $[1,x]$. Under the heuristic assumption that the residues are as likely to be smooth as a randomly chosen integer in $[1,x]$, the probability that a randomly chosen residue v_i is y-smooth is approximately $\psi(x,y)/x \approx u^{-u}$, where $u = \log x/\log y$. The assumption on the residues is necessary, since they form a special subset of the interval $[1,x]$ and may not be as likely to be smooth a randomly chosen integer from the interval. However, as a heuristic, the following principle is obtained: smaller residues lead to faster factorisations.

1.1.2 Finding Smooth Residues

Dixon's random squares method generates residues by randomly choosing integers u from the interval $[1,N]$ and computing $v(u)$. The residues are of size $O(N)$ and are tested individually for smoothness by trial division by primes up to the smoothness bound y. Morrison and Brillhart's continued fractions method employs an idea due to Lehmer and Powers [100] whereby quadratic residue modulo N are computed from the continued fractions expansions of $\sqrt{lN}$, for small square-free $l \in \mathbb{Z}$: if a_i/b_i is the i-th convergent of $\sqrt{lN}$, then $v_i = a_i^2 - lNb_i^2$ is a quadratic residue modulo N. Like Dixon's algorithm, the continued fractions algorithm tests for smoothness by trial division. The residues obtained from the continued fractions expansion of $\sqrt{lN}$ are of size $O(\sqrt{lN})$. Therefore, the continued fractions algorithm generates resides with substantially higher smoothness probabilities than those occurring in Dixon's algorithm.

Based on an idea due to Schroeppel (see [143, Section 6]) for collectively identifying smooth polynomial values, Pomerance introduced the quadratic sieve algorithm. The algorithm generates quadratic resides of size $O(\sqrt{N})$ by evaluating the polynomial $f = x^2 - N$ at x near $\sqrt{N}$. The identification of smooth residues is then based on the observation that $p \mid f(r)$, for some $r \in \mathbb{Z}$, if and only if $p \mid f(r+kp)$, for all $k \in \mathbb{Z}$. In particular, to identify smooth values of f, the quadratic sieve begins by selecting a positive number M and initialising an array containing the values $|f(x)|$, for all $x \in [\sqrt{N} - M, \sqrt{N} + M] \cap \mathbb{Z}$. For each prime $p \leq y$ and each root r of f modulo p, the values $|f(r + kp)|$, where $k \in \mathbb{Z}$, that are contained in the array are retrieved one at a time, divided by the highest power of p that divides them, and the quotient returned to the array. Once this process, called *sieving*, has been completed for all primes $p \leq y$, those values in the array containing 1 correspond to y-smooth values of f. The y-smooth values identified by sieving are then trial divided to obtain their factorisations. Using a sieve to identify smooth values eliminates unnecessary and expensive trial divisions of non-smooth residues. In practice, sieving can be made more efficient by initialising the array with approximations to $\log|f(x)|$ instead of $|f(x)|$ and subtracting integer multiples of $\log p$ from those entries such that $p \mid f(x)$. For this approach, the effect of numerical rounding must be taken into account.

When sieving, the residues produced by f grow as $|x - \sqrt{N}|$ does. If sieving is performed over the interval $[\sqrt{N} - M, \sqrt{N} + M] \cap \mathbb{Z}$, where $M \ll N$, the values taken by $|f(x)|$ for x near the boundary of the interval are of approximate size $2M\sqrt{N}$. Multiple polynomial variants of the quadratic sieve (see [144]) aim to limit the growth of residues by sieving multiple quadratic polynomials over shorter intervals. The most successful of these approaches, due to Montgomery [156], uses quadratic polynomials of the form $f(x) = a^2x^2 + bx + c$, with $\gcd(a, N) = 1$ and $b^2 - 4a^2c \equiv 0 \pmod{N}$. Polynomials of this form are seen to produce quadratic residues modulo N:

$$f(x) = \left(ax + \frac{b}{2a}\right)^2 - \frac{b^2 - 4a^2c}{4a^2} \equiv \left(ax + \frac{b}{2a}\right)^2 \pmod{N}.$$

Therefore, the quadratic sieve can proceed as before by consecutively sieving each polynomial for y-smooth values. However, there is now the advantage that, once the residues produced by a polynomial become too large, smaller residues may be obtained by switching to a new polynomial. Switching polynomials comes at the cost of recomputing the roots of the new polynomial modulo primes up to the smoothness bound. This problem is overcome by another variant of the quadratic sieve called the self initialising quadratic sieve. Details of this algorithm and its development are not given here. Instead, the reader is referred to the literature [148, 137, 6, 39].

The time spent sieving may be reduced by weakening the requirement that the residues be y-smooth, thus increasing the supply of relations. Variants of the quadratic sieve that use this approach, called *large prime* variants, introduce an additional bound y_1 and require that residues be y-smooth with the exception of one, two, or sometimes three prime factors in the interval $[y, y_1]$. Relations that are y-smooth are then called *full-relations* and those containing additional large primes are called *partial relations.* Large prime variants of the quadratic sieve combine partial relations multiplicatively to obtain new residues that are the product of a square and a y-smooth integer. Exponent vectors can then be formed for the y-smooth factors of the resulting residues and linear algebra performed as normal. Overall, the construction of additional exponents vectors from partial relations can be used to significantly reduce the time spent sieving. Details of large prime variants of the quadratic sieve can be found in the literature [105, 24, 111].

1.1.3 Complexity Estimates

The relationship between the time spent generating relations and the running time of an algorithm based on the Morrison–Brillhart approach is clear: the running time of the algorithm is governed by the effort required to find sufficiently many smooth residues as to guarantee the existence of a linear dependence among the exponent vectors. In Section 1.1.1, heuristics were provided to illustrate the dependence of smoothness probabilities on the size of the quadratic residues produced. In this section, the arguments presented there are extended to provide a foundation for analysing the complexity of algorithms based on the Morrison–Brillhart approach. The reader is referred to the discussion by Buhler, Lenstra and Pomerance [29, Section 10] for an in depth treatment of the results stated in this section.

For the analysis of this section, the following function is introduced: for real variables x, u, v with $0 \leq u \leq 1$, the *L-function* $L_x[u, v]$ is defined by

$$L_x[u, v] = \exp\left(v(\log x)^u(\log\log x)^{1-u}\right).$$

As the parameter u varies from 0 to 1, the L-function varies between polynomial and exponential functions of $\log x$.

The Morrison–Brillhart approach requires that as $N \to \infty$, $y^{1+o(1)}$ y-smooth quadratic residues $v(u_i)$ are found. Suppose that some procedure produces residues $v(u_i)$ with absolute value lying in the interval $[1, x]$. If the residues are just as likely to be y-smooth as random integers in $[1, x]$, and can each be tested for smoothness in time $y^{o(1)}$, then the expected effort required to find the required number of y-smooth residues is $xy^{1+o(1)}/\psi(x, y)$. In general, it can not be asserted that the residues are random. However, $xy^{1+o(1)}/\psi(x, y)$ provides a heuristic estimate for the time required to find sufficiently many y-smooth residues $v(u_i)$. With this heuristic, the following theorem due to Buhler et al. [29, Theorem 10.1] bounds the effort required as a function of x:

Theorem 1.1.2. *Let $g(y)$ be a function that is defined for all $y \geq 2$ such that $g(y) \geq 1$ and $g(y) = y^{1+o(1)}$ as $y \to \infty$. Then as $x \to \infty$,*

$$\frac{xg(y)}{\psi(x, y)} \geq L_x[1/2, \sqrt{2} + o(1)]$$

uniformly for all $y \geq 2$. Moreover, equality holds for $x \to \infty$ if and only if $y = L_x[1/2, 1/\sqrt{2} + o(1)]$.

For each of the algorithms discussed in Section 1.1.2, there exists a constant $c > 0$ such that the residues tested for smoothness are bounded in absolute value by $x = N^{1/c+o(1)}$, for $N \to \infty$. By choosing the smoothness parameter y in each algorithm such that

$$y = L_{N^{1/c+o(1)}}[1/2, 1/\sqrt{2} + o(1)] = L_N[1/2, 1/\sqrt{2c} + o(1)], \quad \text{for } N \to \infty,$$

it follows that the residues can each be tested for smoothness in time $y^{o(1)}$ using, for instance, the elliptic curve smoothness test [110]. In the case of the quadratic sieve, residues are tested for smoothness in the same time by sieving. Moreover, Theorem 1.1.2 implies that heuristic estimate $xy^{1+o(1)}/\psi(x, y)$ for the time spent searching for the residues is minimised for this choice of y, with the time equaling

$$L_{N^{1/c+o(1)}}[1/2, \sqrt{2} + o(1)] = L_N[1/2, \sqrt{2/c} + o(1)] = y^{2+o(1)}, \quad \text{for } N \to \infty.$$

If block Wiedemann or block Lanzcos are used for the matrix step, then the time taken is proportional to the product of the dimension and weight of the matrix. The number of nonzero entries in each row is $O(\log(N)) = y^{o(1)}$ and each dimension of the matrix is $y^{1+o(1)}$. Therefore, the matrix step takes time $y^{2+o(1)}$. As a result, the heuristic running time of each algorithm discussed in Section 1.1.2 is the form $L_N[1/2, \sqrt{2/c} + o(1)]$, for some constant $c > 0$ and $N \to \infty$. For the quadratic sieve, $c = 2$, resulting in a heuristic running time of $L_N[1/2, 1 + o(1)]$, for $N \to \infty$. For further examples, see Examples 10.5–10.7 provided by Buhler et al. [29, Section 10].

Each of the algorithms discussed in Section 1.1.2 produce residues of size exponential in $\log N$. The above analysis shows that such an algorithm has a heuristic running time of $L_N[1/2, v+o(1)]$, for some constant $v > 0$. To obtain an algorithm with running time $L_N[u, v + o(1)]$, where $u < 1/2$, the bound x on the numbers tested for smoothness must be at most subexponential in $\log N$. Such a bound is achieved by the number field sieve.

1.2 The Number Field Sieve

The number field sieve [102] was introduced by Pollard [141] in 1988 for the factorisation of integers of the form $x^3 + k$, where x and k are integers such that x is large and k is small. The algorithm was subsequently developed further by Lenstra, Lenstra, Manasse, and Pollard [104] to obtain an algorithm for factoring integers of the form $r^e - s$, where r and $|s|$ are small positive integers and e is large. Their algorithm, now referred to as the special number field sieve, may be applied more generally to the factorisation of integers of the form $ar^e + bs^j$ (see [56]). The designation as special results from the development of the general number field sieve by Buhler, Lenstra and Pomerance [29], which is capable of factoring integers without special form. In this section, a brief description and analysis of the number field sieve is provided. Then the polynomial selection problem for the number field sieve is defined. The section concludes by introducing the definition of smooth elements in number fields and discussing their role in the number field sieve.

In this section and throughout the thesis, results from algebraic number theory are extensively used. For relevant background, the reader is referred to the texts of Marcus [114] and Narkiewicz [128].

1.2.1 Outline of the Number Field Sieve

The number field sieve begins with the selection of two integer polynomials

$$f_1(x) = \sum_{i=0}^{d_1} a_{1,i} x^i \quad \text{and} \quad f_2(x) = \sum_{i=0}^{d_2} a_{2,i} x^i,$$

with $f_1 \neq \pm f_2$, that are primitive and irreducible over $\mathbb{Q}$, and for which there exists an integer m such that $f_i(m) \equiv 0 \pmod{N}$, for $i = 1, 2$. Associated with each polynomial f_i is a number field $K_i = \mathbb{Q}(\alpha_i)$, where $\alpha_i \in \mathbb{C}$ is a root of f_i. Let $\mathbb{Q}_N$ denote the ring of rational numbers with denominator coprime to N. Then the requirement that f_1 and f_2 share m as a root modulo N gives rise to reduction homomorphisms $\phi_i : \mathbb{Q}_N[\alpha_i] \to \mathbb{Z}/N\mathbb{Z}$ induced by $\phi_i(\alpha_i) \equiv m \pmod{N}$, for $i = 1, 2$. In terms of the general algebraic approach to congruence of squares factoring discussed in Section 1.1, the number field sieve may be described by the choice of ring $A = \mathbb{Q}_N[\alpha_1] \times \mathbb{Q}_N[\alpha_2]$ and homomorphism $\varphi : A \to \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ defined by $(\delta_1, \delta_2) \mapsto (\phi_1(\delta_1), \phi_2(\delta_2))$. A congruence of squares modulo N is therefore obtained by finding a square in $\mathbb{Q}_N[\alpha_1] \times \mathbb{Q}_N[\alpha_2]$ with φ-image that lies on the diagonal of $\mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$.

The method used in the number field sieve to construct a square in $\mathbb{Q}_N[\alpha_1] \times \mathbb{Q}_N[\alpha_2]$ involves multiplicatively combining elements of the form $(a - b\alpha_1, a - b\alpha_2)$, for coprime integers a and b. That is, a set $\mathcal{S}$ of coprime integer pairs (a, b) is constructed such that, for $i = 1, 2$,

$$\prod_{(a,b) \in \mathcal{S}} (a - b\alpha_i) = \gamma_i^2, \text{ for some } \gamma_i \in \mathbb{Q}_N[\alpha_i]. \tag{1.1}$$

For all $(a, b) \in \mathbb{Z}^2$, the φ-image of $(a - b\alpha_1, a - b\alpha_2)$ lies on the diagonal of $\mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$. Therefore, after a set $\mathcal{S}$ of coprime integer pairs satisfying (1.1) is constructed by the algorithm, a congruence of squares modulo N is obtained:

$$\phi_1(\gamma_1)^2 \equiv \phi_2(\gamma_2)^2 \pmod{N}.$$

The approach used in the number field sieve to construct a congruence of squares requires that three problems be solved:

1. The problem of constructing the polynomials f_1 and f_2.

2. The problem of finding a set $\mathcal{S}$ of coprime integer pairs satisfying (1.1).

3. Given a set $\mathcal{S}$ satisfying (1.1), the problem of computing the roots γ_1 and γ_2.

In the remainder of this section, the methods used in the number field sieve to solve the first two problems are briefly outlined. The methods used to address the third problem are not discussed in this thesis. Instead, the reader is referred to the literature [29, 44, 19, 120, 54, 129] for details. In Section 1.2.2, a fourth problem is considered: determining the asymptotic complexity of the number field sieve.

Polynomial Selection Briefly

A particularly simple method for generating polynomials for the number field sieve is the *base-m method*. The method, introduced by Buhler et al. [29, Section 3], begins with the selection of a degree parameter $d \geq 2$ (selection of d is discussed in Section 1.2.2). After setting $m = \lfloor N^{1/d} \rfloor$, N is then written in base-m:

$$N = a_d m^d + a_{d-1} m^{d-1} + \ldots + a_1 m + a_0,$$

where the coefficients $a_0, \ldots, a_d \in [0, m) \cap \mathbb{Z}$. Finally, the polynomials $f_1 = \sum_{i=0}^{d} a_i x^i$ and $f_2 = x - m$, with common root m modulo N, are obtained. For $N > 2^{d^2}$, the polynomial f_1 is monic [29, Proposition 3.2] (in fact, $a_d = 1$ whenever $N > 1.5(d/\log 2)^d$ [49, Exercise 6.8]) and is therefore primitive. However, the polynomial f_1 may have a nontrivial factorisation $f_1 = gh$. In this case, a result (implicitly) obtained by Brillhart, Filaseta, Odlyzko [26] implies that $g(m)h(m) = N$ is a nontrivial factorisation of N whenever $m \geq 3$ (see also [49, Exercise 6.9]). Therefore, f_1 is either irreducible over $\mathbb{Q}$ or can be factored in time polynomial in $\log N$ (see [103]) and a factorisation of N obtained.

Polynomial selection for the special number field sieve exploits the special form of N to produce the polynomials f_1 and f_2. Suppose that $N = r^e - s$, where r and $|s|$ are small positive integers and e is large. Then polynomial selection proceeds by choosing a degree parameter d and letting k be the least positive integer such that $k \cdot d \geq e$. For $t = s \cdot r^{k \cdot d - e}$ and $m = r^k$, the polynomials $f_1 = x^d - t$ and $f_2 = x - m$ satisfy $f_i(m) \equiv 0 \pmod{N}$, for $i = 1, 2$. Lenstra et al. [104, Section 2.5] describe simple criteria that may be used to determine if f_1 is irreducible. If f_1 is not irreducible, either a factor of N is found, or f_1 can be replaced by one of its irreducible factors. For improvements to this method and details on polynomial selection for the larger class of integers of the form $N = ar^e + bs^j$, the reader is referred to [56].

A method of specialised polynomial selection also exists for N of the form $\sum_{i=1}^{w} b_i 2^{c_i}$, with $b_1, \ldots, b_w \in \{-1, 1\}$ and w small. Numbers of this from are said to have *low weight* and polynomial selection may be performed by the algorithm described by Schirokauer [151, Algorithm 2.1].

Constructing Squares in Number Fields

The construction of a nonzero square in $\mathbb{Q}_N[\alpha_1] \times \mathbb{Q}_N[\alpha_2]$ is, by a large margin, the most time consuming and challenging stage of the number field sieve algorithm. In the number field sieve, this problem is approached by multiplicatively combining elements of the form $(a - b\alpha_1, a - b\alpha_2)$, where a and b are coprime integers. The special form of the elements $(a - b\alpha_1, a - b\alpha_2)$ then permits the use of ideas from the Morrison–Brillhart approach and the quadratic sieve algorithm, such as using sieves to efficiently identify smooth polynomial values and combining relations using linear algebra.

Let $\sigma_{i,1}, \ldots, \sigma_{i,d_i}$ denote the field embeddings of K_i in the field $\mathbb{C}$, for $i = 1, 2$. Then the *norm map* $N_{K_i} : K_i \to \mathbb{Q}$ is defined by $N_{K_i}(x) = \prod_{j=1}^{d_i} \sigma_{i,j}(x)$, for all $x \in K_i$. It follows immediately from the definition that the norm map is multiplicative. As a result, if $\prod_{(a,b)\in\mathcal{S}}(a - b\alpha_i)$ is a square in K_i, then its norm $\prod_{(a,b)\in\mathcal{S}} N_{K_i}(a - b\alpha_i)$ must be a square in $\mathbb{Q}$. Define homogeneous polynomials $F_i(x, y) = f_i(x/y) \cdot y^{d_i}$, for $i = 1, 2$. Then, for all $(a, b) \in \mathbb{Z}^2$,

$$N_{K_i}(a - b\alpha_i) = \prod_{j=1}^{d_i} (a - b\sigma_{i,j}(\alpha_i)) = b^{d_i} \prod_{j=1}^{d_i} \left(ab^{-1} - \sigma_{i,j}(\alpha_i)\right) = a_{d_i}^{-1} F_i(a, b). \tag{1.2}$$

Therefore, if $|\mathcal{S}|$ is even and $\prod_{(a,b)\in\mathcal{S}} F_i(a, b)$ is a square in $\mathbb{Z}$, then $\prod_{(a,b)\in\mathcal{S}} N_{K_i}(a - b\alpha_i)$ is a square in $\mathbb{Q}$. In the number field sieve, such a set $\mathcal{S}$ is constructed by selecting smoothness bounds $y_1, y_2 > 0$ and using sieving to identify coprime integer pairs (a, b) such that $F_i(a, b)$ is y_i-smooth, for $i = 1, 2$. Each pair with this property is called a *relation*. For each relation, a corresponding exponent vector in an $\mathbb{F}_2$-vector space is created from the prime factorisations of $F_1(a, b)$ and $F_2(a, b)$. Then given sufficiently many relations, linear algebra over $\mathbb{F}_2$ is used to find a subset $\mathcal{S}$ such that $\prod_{(a,b)\in\mathcal{S}} F_i(a, b)$ is a square in $\mathbb{Z}$, for $i = 1, 2$. To ensure that $|\mathcal{S}|$ is even, an extra entry containing 1 is appended to each exponent vector. Thus any linearly dependent subset of the exponent vectors must contain an even number of vectors.

Remark 1.2.1. Throughout the thesis, given a polynomial $f \in \mathbb{Z}[x]$, upper case F is used to denote the homogeneous polynomial $F(x, y) = f(x/y) \cdot y^{\deg f}$. Similar notation is also used for the remaining letters of the alphabet.

Although necessary for x to be a square in a number field K, the requirement that $N_K(x)$ is a rational square is far from sufficient. For example, if $K = \mathbb{Q}(\sqrt{2})$, then $N_K(3 \pm \sqrt{2}) = 7$. Therefore the product $N_K(3 + \sqrt{2})N_K(3 - \sqrt{2}) = 7^2$ is a square in $\mathbb{Q}$. However, the product $(3 + \sqrt{2})(3 - \sqrt{2}) = 7$ is certainly not a square in K. In order to describe why the norm being a square is not sufficient, some notation must first be introduced. Throughout the thesis, the ring of algebraic integers in a number field K is denoted by $\mathcal{O}_K$. The abelian group of fractional ideals of $\mathcal{O}_K$ is denoted by $\mathcal{I}_K$. It is well-known that each ideal in $\mathcal{I}_K$ factors uniquely (up to order) into a product of prime ideals of $\mathcal{O}_K$. An ideal $\mathfrak{a} \in \mathcal{I}_K$ with prime ideal factorisation $\mathfrak{a} = \mathfrak{p}_1^{\varepsilon_1} \cdots \mathfrak{p}_n^{\varepsilon_n}$ is said to be a square if and only if each ε_i is even. If $x \in K^\times$ is a square in K, then the principal fractional ideal it generates, denoted $\langle x \rangle$, is a square in $\mathcal{I}_K$. Each prime ideal in the factorisation of $\langle x \rangle$ contributes a prime power to $N_K(x)$. The norm of an element combines these factors together therefore losing information about which primes contributed to the norm. As a result, the norm of an element $x \in K^\times$ may be a square in $\mathbb{Q}$ without $\langle x \rangle$ being a square in $\mathcal{I}_K$. For example, the ring of integers in $K = \mathbb{Q}(\sqrt{2})$ is $\mathbb{Z}[\sqrt{2}]$. The ideal generated by 7 in

$\mathbb{Z}[\sqrt{2}]$ factors as the product of prime ideals $\mathfrak{p}_1 = \langle 3+\sqrt{2}\rangle$ and $\mathfrak{p}_2 = \langle 3-\sqrt{2}\rangle$. Each of the ideals $\mathfrak{p}_1$ and $\mathfrak{p}_2$ contributes a factor of 7 to $N_K(7)$. Therefore, $N_K(7) = 7^2$ is a square in $\mathbb{Q}$, whereas the ideal $\langle 7\rangle = \mathfrak{p}_1\mathfrak{p}_2$ is not a square in $\mathcal{I}_K$.

To address the problem of the norm failing to be sufficient to determine if a square has been found, the exponent vector created in the number field sieve for a relation (a,b) almost completely separates the contributions of the prime ideals in the factorisation of $\langle a - b\alpha_i\rangle$. However, for $x \in K^\times$, the principal ideal $\langle x\rangle$ may be a square in $\mathcal{I}_K$, while x is not a square in K. For example, in $\mathbb{Z}[\sqrt{2}]$ the associates $\pm(3-\sqrt{2})$ have the same prime ideal factorisation. Therefore, $\langle 3-\sqrt{2}\rangle\langle -3+\sqrt{2}\rangle = \langle 3-\sqrt{2}\rangle^2$ is the square of an ideal. However, the product $(3-\sqrt{2})(-3+\sqrt{2}) = -1\cdot(3-\sqrt{2})^2$ is not a square in $\mathbb{Q}(\sqrt{2})$. This example demonstrates an obstruction associated with units. Unfortunately, this is not the sole obstruction that prevents $\langle x\rangle$ being a square in $\mathcal{I}_K$ from implying that x is a square in K.

A detailed description of the obstructions is provided by Buhler et al. [29, Section 6]. There it is shown that the obstructions may be described by a $\mathbb{F}_2$-vector space of low dimension. In particular, if the polynomials are generated with the base-m method, $d \geq 2$, and $N > d^{2d^2}$, then the dimension of the vector space is less than $\log N/\log 2$. Adleman [3] described how to construct maps $\chi_{i,1},\ldots,\chi_{i,n_i}$, called *quadratic characters*, from a subset of $\mathbb{Z}[\alpha_i]$ to the multiplicative group $\{-1,1\}$ such that the condition $\prod_{(a,b)\in\mathcal{S}}\chi_{i,j}(a-b\alpha_i) = 1$, for $1 \leq j \leq n_i$, is necessary for $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha_i)$ to be a square in K_i. Moreover, the quadratic characters have the property that if n_i is large when compared to the dimension of the $\mathbb{F}_2$-vector space describing the obstructions, then it is highly likely that the condition is also sufficient (see [29, Section 8]). By identifying $\{-1,1\}$ with the additive group $\{0,1\}$, entries corresponding to $\chi_{i,j}(a-b\alpha_i)$ are appended to the exponent vector of each relation (a,b). Then a linearly dependent subset of the exponent vectors leads to a set $\mathcal{S}$ of relations such that $\prod_{(a,b)\in\mathcal{S}}(a-b\alpha_i)$ is very likely a square in K_i, for $i = 1,2$.

Summary of the Algorithm

The number field sieve can be summarised by the following steps:

Polynomial Selection: The selection of two integers polynomials f_1 and f_2 with $f_1 \neq \pm f_2$, that are primitive and irreducible over $\mathbb{Q}$, and for which there exists an integer m such that $f_i(m) \equiv 0 \pmod{N}$, for $i = 1,2$.

Sieving: The identification of relations by means of a sieve. Relations are coprime integer pairs (a,b) such that $F_i(a,b)$ is y_i-smooth, for $i = 1,2$. The smoothness bounds y_1 and y_2 are parameters of the algorithm. For large prime variants of the number field sieve, the sieve step is also used to identify partial relations: coprime integer pairs (a,b) such that $F_i(a,b)$ is y_i-smooth with the exception of a small number of large primes, for $i = 1,2$.

Linear Algebra: Exponent vectors are created for the relations found in the sieve step and linear algebra over $\mathbb{F}_2$ used to find a set $\mathcal{S}$ of relations that satisfy (1.1). In practice, multiple sets of relations satisfying (1.1) are usually found.

Square Root: For each set $\mathcal{S}$, the roots γ_1 and γ_2 in (1.1) are computed.

On the completion of all steps of the algorithm, each set $\mathcal{S}$ yields a congruence of squares modulo N: $\phi_1(\gamma_1)^2 \equiv \phi_2(\gamma_2)^2 \pmod{N}$. Then a factor of N is potentially found by computing $\gcd(\phi_1(\gamma_1) \pm \phi_2(\gamma_2), N)$. If this fails to return a factor for each set $\mathcal{S}$, the algorithm returns to the sieve step and gathers more relations. Then the linear algebra and square root steps are repeated.

1.2.2 Complexity Estimates

Algorithms based on the Morrison–Brillhart approach that test numbers of size $N^{\Theta(1)}$ for smoothness, have heuristic running times that are restricted to the form $L_N[1/2, v + o(1)]$, for constants $v > 0$, and $N \to \infty$. The number field sieve greatly improves over previous algorithms based on the Morrison–Brillhart approach, such as those mentioned in Section 1.1.2, by only requiring numbers of size $N^{o(1)}$ to be tested for smoothness. The resulting increase in smoothness probabilities leads to the number field sieve having a conjectured running time [29, Conjecture 11.2], under an optimal choice of parameters, of

$$L_N\left[1/3, (64/9)^{1/3} + o(1)\right], \quad \text{for } N \to \infty. \tag{1.3}$$

In this section, the choices of parameters leading to this estimate are briefly reviewed. The reader is referred to the original analysis of Buhler et al. [29] for an in-depth treatment of the results stated in this section.

For the analysis, it is assumed that the polynomials f_1 and f_2 are generated with the base-m method. Denote by d the degree of the nonlinear polynomial. The smoothness bounds are assumed to be equal with $y = y_i$, for $i = 1, 2$. Finally, it is assumed that sieving is performed over all pairs (a, b) contained in the region $\mathcal{A} = [-u, u] \times [1, u]$, where $u \geq 1$ is a parameter to be chosen later. The choice of polynomials guarantees that the coefficients of $F_1(x, y)$ and $F_2(x, y)$ are bounded by $m \leq N^{1/d}$. Therefore, the values taken by $F_1(a, b)F_2(a, b)$, for $(a, b) \in \mathcal{A}$, are bounded in absolute value by

$$(u + um)(d + 1)mu^d \leq 2dm^2u^{d+1} \leq 2dN^{\frac{2}{d}}u^{d+1}.$$

Hence, the numbers tested for smoothness in the sieve stage of the algorithm are bounded by

$$x = 2dN^{\frac{2}{d}}u^{d+1}. \tag{1.4}$$

Throughout this section, all $o(1)$ terms are for $N \to \infty$. The dimension of the exponent vectors, and thus the number of relations required, is $y^{1+o(1)}$. Therefore, under the assumption that the values $F_1(a, b)F_2(a, b)$ are just as likely to be smooth as an integer chosen at random from the interval $[1, x]$, it follows from Theorem 1.1.2 that the time spent sieving is minimised for $y = L_x[1/2, 1/\sqrt{2} + o(1)]$. Moreover, the expected number of coprime integer pairs $(a, b) \in \mathcal{A}$ that are required to be tested for smoothness in order for $y^{1+o(1)}$ relations to be found is $L_x[1/2, \sqrt{2} + o(1)]$. The number of coprime integer pairs $(a, b) \in \mathcal{A}$ is approximately $12u^2/\pi^2$, where the factor of $6/\pi^2$ takes into account the (asymptotic) probability that a and b are coprime (see [9, Theorem 3.9]). Therefore, u should satisfy

$$u^2 \geq L_x[1/2, \sqrt{2} + o(1)].$$

Here the factor of $12/\pi^2$ has been absorbed into the $o(1)$ term on the right hand side. Taking logarithms and squaring gives

$$2\log^2 u \geq (1 + o(1)) \log x \log\log x.$$

Since $t/\log t$ is increasing for $t \geq e$, it follows that u should satisfy

$$\frac{\log^2 u}{\log\log u} \geq (1 + o(1)) \log x \geq (1 + o(1)) \left(\frac{2}{d} \log N + (d + 1) \log u\right).$$

Applying a technical lemma of Buhler et al. [29, Lemma 10.9], with $k \geq (1 + o(1))(d + 1)$ and $l \geq (2 + o(1)) \log N^{1/d}$, results in the lower bound

$$u \geq \exp\left[\left(\frac{1}{2} + o(1)\right)\left(d\log d + \sqrt{(d\log d)^2 + 4\log(N^{1/d})\log\log(N^{1/d})}\right)\right]. \tag{1.5}$$

Since $y = L_x[1/2, 1/\sqrt{2} + o(1)]$, it follows that y satisfies the same lower bound. Buhler et al. [29, Section 11] showed that the time taken by sieve step is $u^{2+o(1)}$, whilst the time taken by the matrix and square root steps is $y^{2+o(1)}$. Therefore, setting u and y equal to the right hand side of (1.5), the (conjectured) asymptotic running time of the number field sieve, for fixed d, is

$$\exp\left[(1 + o(1))\left(d\log d + \sqrt{(d\log d)^2 + 4\log(N^{1/d})\log\log(N^{1/d})}\right)\right]. \tag{1.6}$$

Choosing d to minimise this expression results in the optimal choice of

$$d = \left(3^{1/3} + o(1)\right)\left(\frac{\log N}{\log\log N}\right)^{1/3}. \tag{1.7}$$

Then substituting (1.7) into (1.6) leads to the conjectured running time (1.3).

With values of y, u and d used to establish the running time, the bound (1.4) on the size of the numbers tested for smoothness in the sieve stage becomes

$$x = L_N\left[2/3, (64/3)^{1/3} + o(1)\right].$$

As a result, the values tested for smoothness in the number field sieve are asymptotically smaller than those appearing in the quadratic sieve.

For fixed d, (1.6) simplifies to

$$L_N\left[1/2, \sqrt{4/d} + o(1)\right], \quad \text{for } N \to \infty.$$

Recall that the running time of the quadratic sieve is $L_N[1/2, 1 + o(1)]$, for $N \to \infty$. This suggests that the number field sieve will surpass the quadratic sieve only when $d \geq 5$. By substituting $d = 5$ into (1.7) and naively ignoring the $o(1)$ term, solving for N suggests that the crossover point, where the number field sieve will surpass the quadratic sieve, is somewhere around 100 decimal digits.

1.2.3 The Polynomial Selection Problem

Asymptotically, the number field sieve obtains a greater number of smooth values over previous algorithms as a result of the reduction in size from exponential to subexponential of the values tested for smoothness. The complexity analysis of Section 1.2.2 used base-m polynomials to demonstrate this advantage over previous algorithms. Considered asymptotically and for general N, the base-m method is, in a weak sense, optimal [29, Section 12.10] (see also Section 2.2.4). However, in practice, N is fixed rather than tending to infinity. Furthermore, $o(1)$ terms in the analysis of the number field sieve conceal the true influence of a particular choice of polynomials. Therefore, in practice, improvements to the base-m method are possible.

The polynomial selection problem is concerned with determining a choice of polynomials that guarantees the best practical performance. For a given N, the *polynomial selection problem* is to find a pair of integer polynomials f_1 and f_2 with the following properties:

Structural: The polynomials satisfy the conditions necessary for use with the number field sieve: $f_1 \neq \pm f_2$; f_1 and f_2 are both primitive and irreducible over $\mathbb{Q}$; there exists a known integer m such that $f_i(m) \equiv 0 \pmod{N}$, for $i = 1, 2$.

High yield: The associated homogeneous polynomials $F_1(x, y)$ and $F_2(x, y)$ yield many smooth values thus reducing the time spent sieving.

Throughout the thesis, polynomials that satisfy the required structural properties are referred to as *number field sieve polynomials.* Consequently, the polynomial selection problem requires that a pair of number field sieve polynomials, that maximise yield, be found.

The polynomial selection problem may be viewed as being comprised of two parts: polynomial generation and ranking. The polynomial generation problem is concerned with constructing good number field polynomials, while the polynomial ranking problem is concerned with determining the best polynomials among many given pairs. Both the generation and ranking problems require an understanding of the properties which influence polynomial yield. Due largely to the work of Brent, Murphy and

Montgomery [124, 125, 126], these properties and their influence are well understood. Moreover, out of that work evolved effective methods for ranking number field sieve polynomials.

Polynomial generation for the special number field sieve is well understood [56]. In contrast, polynomial generation for numbers without special form remains an area of active research. The efforts of research on this problem have focused on two different approaches. The first and most successful approach, is to modify or extend the base-m method. Algorithms based on this approach [126, 91, 90] have been used in a string of record factorisations [45, 33, 32, 11, 92] and remain the state of the art. The second approach, employs techniques from the algorithmic geometry of numbers to produce nonlinear polynomials of equal degree. Algorithms based on this approach remain the best known for numbers up to approximately 120 digits. Examples of numbers factored using nonlinear polynomials can be found in [54].

In this thesis, attention is focused primarily on the polynomial generation problem for numbers without special form. The contributions to this problem are twofold: existing methods for polynomial generation are generalised and a new approach to polynomial generation is developed.

1.2.4 Smooth Elements in Number Fields

The concept of a y-smooth integer may be naturally generalised to number fields:

Definition 1.2.2. Let K be a number field. A nonzero element $x \in \mathcal{O}_K$ is called *y-smooth* if $N_K(x)$ is a y-smooth integer.

An element $x \in \mathcal{O}_K$ is y-smooth if for every prime ideal $\mathfrak{p} \subset \mathcal{O}_K$, $\mathfrak{p} \mid \langle x \rangle$ implies that $\mathfrak{p} \cap \mathbb{Z} = \langle p \rangle$ for a rational prime $p \leq y$. Therefore, the smoothness of an element $x \in \mathcal{O}_K$ describes how the principal ideal that it generates factors over prime ideals. For each rational prime p, let $\nu_p : \mathbb{Q} \to \mathbb{Z} \cup \{\infty\}$ denote the *p-adic valuation*, which is defined as follows: for nonzero $x \in \mathbb{Q}$, $\nu_p(x)$ is the unique integer v such that $x = p^v(s/t)$, where s and t are integers not divisible by p; and $\nu_p(0) = \infty$. The concept of smoothness may be extended to entire of $K^\times$ by defining $x \in K^\times$ to be y-smooth if $\nu_p(N_K(x)) \neq 0$ implies that $p \leq y$. In the special case $K = \mathbb{Q}$, a quotient $u/v \in \mathbb{Q}$ of coprime integers u and v is y-smooth if and only if uv is y-smooth. The smoothness of an element $x \in K$ describes how the principal fractional ideal that it generates factors over prime ideals.

Smooth elements in number fields play a role in the number field sieve whenever one of the polynomials f_1 or f_2 is monic: if f_i is monic, then (1.2) implies that each relation gives rise to a y_i-smooth element in K_i. More generally, relations give rise to y_i-smooth elements in K_i whenever the leading coefficient of f_i is y_i-smooth. In practice, it is beneficial to force the polynomials f_1 and f_2 to satisfy this property [126, Section 5.1.1]. In addition to the number field sieve, smooth elements in number fields appear in algorithms for computing discrete logarithms in finite fields [64, 86], finding solutions to the Pell equation [109] and computing class groups [28].

1.3 Outline of the Thesis

Chapter 2 introduces preliminaries on the polynomial selection problem relevant to Chapters 3 and 4 of the thesis. In addition, existing literature on the problem is surveyed.

Chapters 3 and 4 contain new algorithms for generating good number field sieve polynomials. Chapter 3 focuses on lattice-based algorithms for nonlinear polynomial generation. There existing algorithms are reviewed and generalised. In Chapter 4, an entirely new approach to polynomial generation is developed. The development is driven by newly obtained results on the divisibility properties of univariate resultants. An initial realisation of the approach is provided and analysed. In addition, bounds on the existence of number field sieve polynomials with favourable properties are derived using algorithmic and combinatorial methods. Finally, possible improvements and generalisations of the new approach are discussed.

In Chapter 5, the problem of finding smooth algebraic integers in a number field is reformulated as a decoding problem for a family of error-correcting codes called NF-codes. The first algorithm for solving the weighted list decoding problem for NF-codes is provided. Then the algorithm is used to find algebraic integers with norm containing a large smooth factor. Finally, bounds on the existence of such elements are derived using algorithmic and combinatorial methods.

Chapter 6 contains conclusions and suggestions for future avenues of research.

Chapter 2

Preliminaries on Polynomial Selection

This chapter introduces background material on the polynomial selection problem relevant to Chapter 3 and Chapter 4 of the thesis. Existing literature on the problem is also surveyed. Where appropriate, certain topics such as polynomial resultants and lattices are only briefly touched upon, with formal treatments deferred until they are required.

In Chapter 1, it was noted that polynomial selection problem may be considered as being comprised of two parts: polynomial generation and ranking. Accordingly, this chapter is also comprised of two parts. The first part, Section 2.1, begins by examining the properties which influence polynomial yield and methods for their quantification. Then methods for ranking number field sieve polynomials according to their yield are discussed. The second part, Section 2.2, reviews current methods for generating number field sieve polynomials.

2.1 Quantifying Properties which Influence Polynomial Yield

There are two main factors which influence the yield of a number field sieve polynomial. The first of these two factors, called a polynomial's *size properties*, refers to the magnitude of the values taken by the polynomial over the sieve region. The influence of size properties was demonstrated in Section 1.2.2, where the number field sieve's asymptotic advantage over previous algorithms based on the Morrison–Brillhart approach was shown to result from a reduction, from exponential to subexponential, of the size of the values tested for smoothness. The second factor which influences yield, called a polynomial's *root properties*, refers to the distribution of its roots modulo small prime powers. This factor was excluded from the asymptotic considerations of Section 1.2.2, since its effect is hidden within the $o(1)$ terms. However, due largely to the work of Brent, Montgomery and Murphy [124, 125, 126], it is known that in practice, a polynomial with many roots modulo small prime powers produces values that behave, with regard to smoothness probabilities, as if they are much smaller than their true size. Therefore, polynomials with good root properties experience an increased likelihood of producing smooth values on average.

In this section, methods for quantifying the size and root properties number field sieve polynomials are reviewed. Section 2.1.1 begins by introducing polynomial norms used to measure a polynomial's size properties. Then a lower bound on the coefficient size of pairs of number field sieve polynomials is derived. In Section 2.1.2, quantifying the root properties of number field sieve polynomials is considered. There the influence of root properties is discussed further. Finally, Section 2.1.3 reviews approximate methods for ranking polynomial pairs according to their yield.

2.1.1 Quantifying Size Properties: Skewed Polynomial Norms

In this section, polynomial norms used throughout the thesis to measure size properties are introduced. Then a lower bound on the coefficient size of polynomial pairs with a common root modulo N is derived. Throughout, it is assumed that sieving is used to identify all relations contained in a region $\mathcal{A}$ of the form $\mathcal{A} = [-A, A] \times [0, B]$. The actual form of the region depends on the method of sieving. Furthermore, it is known that a rectangular sieve region is not optimal in general [157]. The area of $\mathcal{A}$ is approximately determined by the size of the input N. Therefore, it is assumed that the region's area is fixed. Consequently, $\mathcal{A}$ is determined by the parameter $s = A/B$, called the *skew* of the region.

Skewed Polynomial Norms

Given two polynomials f_1 and f_2, the size of the values taken by their respective homogenisations F_1 and F_2 over the sieve region $\mathcal{A}$ can be roughly quantified by the integral

$$\int_{\mathcal{A}} |F_1(x,y)F_2(x,y)|\, dxdy.$$

Using Hölder's inequality to bound this integral suggests that some indication of the size properties of a degree d polynomial f can be obtained by considering the integral

$$\int_{\mathcal{A}} F(x,y)^2 dxdy = (AB)^{d+1} \cdot \int_0^1 \int_{-1}^1 \left(f\left(\frac{sx}{y}\right) \cdot \left(\frac{y}{\sqrt{s}}\right)^d \right)^2 dxdy. \tag{2.1}$$

The integrand on the right motivates the following choice of coefficient norms:

Definition 2.1.1. Let $f = \sum_{i=0}^d a_i x^i \in \mathbb{R}[x]$ be a degree d polynomial and s a positive real number. For any real number $p \geq 1$, the *skewed p-norm* of f is defined by

$$\|f\|_{p,s} = \left(\sum_{i=0}^d \left| a_i s^{i-\frac{d}{2}} \right|^p \right)^{\frac{1}{p}}.$$

Furthermore, the *skewed ∞-norm* of f is defined by

$$\|f\|_{\infty,s} = \max_i \left| a_i s^{i-\frac{d}{2}} \right| .$$

If $s = 1$, then $\|f\|_{p,s}$ is simply called the *p-norm* of f and denoted by $\|f\|_p$, for real $p \geq 1$ and $p = \infty$.

The skewed ∞-norm is also referred to in the literature as the *sup-norm* by Kleinjung [91].

The norms introduced in Definition 2.1.1 may be used to measure the coefficient size of a number field sieve polynomial. However, they do not consider the size of the values taken by a polynomial over the sieve region, thus ignoring the main factor used to determine the (asymptotic) yield of number field sieve polynomials in the complexity analysis of Section 1.2.2. Therefore, the skewed p-norms only provide a coarse measure of size properties. A finer measure of size properties can be obtained by considering the entire integral (2.1) rather than just the integrand. In particular, the right-hand integral of (2.1), in combination with the assumption that the area of $\mathcal{A}$ is fixed, suggests the following choice of norms:

Definition 2.1.2. Let $f \in \mathbb{R}[x]$ be a degree d polynomial. For a given positive real number s, the *skewed L^2-norm* of f is defined by

$$\|f\|_{L^2,s} = \sqrt{\int_0^1 \int_{-1}^1 \left(f\left(\frac{sx}{y}\right) \cdot \left(\frac{y}{\sqrt{s}}\right)^d \right)^2 dxdy.}$$

Suppose that two nonzero polynomials $f_1, f_2 \in \mathbb{Z}[x]$, both of degree d, are to be sieved over respective regions $\mathcal{A}_1$ and $\mathcal{A}_2$ with equal area. Then (2.1) implies that

$$\frac{\int_{\mathcal{A}_1} F_1(x,y)^2 dxdy}{\int_{\mathcal{A}_2} F_2(x,y)^2 dxdy} = \frac{\|f_1\|^2_{L^2,s_1}}{\|f_2\|^2_{L^2,s_2}},$$

where s_i denotes the skew of $\mathcal{A}_i$, for $i = 1, 2$. Therefore, the measure of size properties provided by the skewed L^2-norm is sufficiently fine as to allow the comparison of number field sieve polynomials. Accordingly, a *skew* of a polynomial $f \in \mathbb{R}[x]$, is defined to be any value $s > 0$ such that $\|f\|_{L^2,s}$ is minimal.

The Resultant Bound

For nonzero coprime polynomials $f_1, f_2 \in \mathbb{Z}[x]$ with a common root modulo N, the *resultant bound* provides a lower bound on the 2-norms of f_1 and f_2:

$$\|f_1\|_2^{\deg f_2} \cdot \|f_2\|_2^{\deg f_1} \geq N. \tag{2.2}$$

The 2-norm may greatly overestimate the coefficient size of polynomials with large skew. To provide tighter bounds, a generalisation of inequality (2.2) is now derived for the skewed 2-norm. To begin, the definition and some properties of the resultant of two polynomials must first be introduced.

Let $f = \sum_{i=0}^{m} a_i x^i$ and $g = \sum_{i=0}^{n} b_i x^i$ be non-constant polynomials with complex coefficients and $a_m, b_n \neq 0$. The *Sylvester matrix* of f and g, denoted $\operatorname{Syl}(f,g)$, is the $(m+n) \times (m+n)$ matrix

$$\operatorname{Syl}(f,g) = \begin{pmatrix} a_m & a_{m-1} & \cdots & \cdots & a_0 & & & \\ & a_m & a_{m-1} & \cdots & \cdots & a_0 & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & a_m & a_{m-1} & \cdots & \cdots & a_0 \\ b_n & b_{n-1} & \cdots & \cdots & b_0 & & & \\ & b_n & b_{n-1} & \cdots & \cdots & b_0 & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & b_n & b_{n-1} & \cdots & \cdots & b_0 \end{pmatrix}$$

where there are n rows containing the a_i, m rows containing the b_i, and all empty entries are 0. The *resultant* of f and g, denoted $\operatorname{Res}(f,g)$, is equal to the determinant of the Sylvester matrix $\operatorname{Syl}(f,g)$. The following two well-known properties of resultants are used to generalise the resultant bound:

- If $\alpha_1, \ldots, \alpha_m \in \mathbb{C}$ are the roots of f, and $\beta_1, \ldots, \beta_n \in \mathbb{C}$ the roots of g, then

$$\operatorname{Res}(f,g) = a_m^n b_n^m \prod_{i,j} (\alpha_i - \beta_j). \tag{2.3}$$

- If $f, g \in \mathbb{Z}[x]$, then $\operatorname{Res}(f,g)$ belongs to the ideal $\langle f, g \rangle \cap \mathbb{Z}$.

Proofs of these two properties can be found in [50, Section 1.3.2]. The two properties imply that coprime non-constant polynomials $f_1, f_2 \in \mathbb{Z}[x]$ with a common root modulo N must satisfy $N \leq |\operatorname{Res}(f_1, f_2)|$. The resultant bound (2.2) is then obtained by using Hadamard's inequality [72] to bound the absolute value of $\det \operatorname{Syl}(f_1, f_2)$. The following lemma generalises the upper bound obtained from Hadamard's inequality:

Lemma 2.1.3. Let $f = \sum_{i=0}^{m} a_i x^i$ and $g = \sum_{i=0}^{n} b_i x^i$ be non-constant polynomials with complex coefficients and $a_m, b_n \neq 0$. Then $|\operatorname{Res}(f,g)| \leq \|f\|_{2,s}^{n} \cdot \|g\|_{2,s}^{m}$, for all $s > 0$. If $m \neq n$, then the inequality is strict, for all $s > 0$.

Proof. Let $\alpha_1, \ldots, \alpha_m$ be the roots of f, and $\beta_1, \ldots, \beta_n$ the roots of g. For $s > 0$,

$$\operatorname{Res}(f,g) = a_m^n b_n^m \prod_{i,j} (\alpha_i - \beta_j) = \left(a_m s^{\frac{m}{2}}\right)^n \left(b_n s^{\frac{n}{2}}\right)^m \prod_{i,j} \left(\frac{\alpha_i}{s} - \frac{\beta_j}{s}\right) = \operatorname{Res}\left(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx)\right).$$

Hence,

$$\big|\mathrm{Res}\,(f,g)\big| = \big|\det \mathrm{Syl}\big(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx)\big)\big| \leq \big\|s^{-\frac{m}{2}} f(sx)\big\|_2^n \cdot \big\|s^{-\frac{n}{2}} g(sx)\big\|_2^m = \big\|f\big\|_{2,s}^n \cdot \big\|g\big\|_{2,s}^m, \quad (2.4)$$

for all $s > 0$, where the inequality is obtained by applying Hadamard's inequality.

To prove the second assertion of the lemma, suppose for contradiction that $m \neq n$ and $|\mathrm{Res}(f,g)| = \|f\|_{2,s}^n \cdot \|g\|_{2,s}^m$, for some value of $s > 0$. Then (2.4) implies that the determinant of the Sylvester matrix $\mathrm{Syl}(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx))$ attains Hadamard's bound, and thus its row vectors are orthogonal. If $m < n$, then the inner product of the first and $(m+1)$th row of $\mathrm{Syl}(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx))$ is $a_m a_0$. Thus, $a_0 = 0$, since a_m is nonzero. Similarly, the inner product of the first and mth row of $\mathrm{Syl}(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx))$ is $a_m a_1 s + a_{m-1} a_0 = a_m a_1 s$, and thus $a_1 = 0$. Continuing in this manner, it follows that if $m < n$, then $f = a_m x^m$. Similarly, if $m > n$, then computing the inner product of the $(n+1)$th and kth row of $\mathrm{Syl}(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx))$, for $n+2 \leq k \leq 2n+1$, shows that $g = b_n x^n$. Therefore, in either case, the inner product of the first and $(n+1)$th row of $\mathrm{Syl}(s^{-\frac{m}{2}} f(sx), s^{-\frac{n}{2}} g(sx))$ is $s^{(m+n)/2} a_m b_n \neq 0$, which contradicts orthogonality. □

Corollary 2.1.4. Let N be a positive integer and $f_1, f_2 \in \mathbb{Z}[x]$ be non-constant coprime polynomials with a common root modulo N. Then $\|f_1\|_{2,s}^{\deg f_2} \cdot \|f_2\|_{2,s}^{\deg f_1} \geq N$, for all $s > 0$. If $\deg f_1 \neq \deg f_2$, then the inequality is strict, for all $s > 0$.

Remark 2.1.5. The bound in Lemma 2.1.3 is attained, for $s > 0$ and $d \geq 1$, by polynomials $f_1 = x^d - s^d$ and $f_2 = x^d + s^d$: the product formula (2.3) implies that $\mathrm{Res}(f_1, f_2) = (2s^d)^d$; and $\|f_1\|_2^d \cdot \|f_2\|_2^d = (s^{d/2}\sqrt{2})^{2d} = (2s^d)^d$. If $d = 1$ and s is an integer, then the lower bound in Corollary 2.1.4 is also attained, since $x - s$ and $x + s$ have a common root modulo $2s$.

The complexity of the number field sieve is largely determined by the size of N and the degree sum $\deg f_1 + \deg f_2$ of the polynomials used (see Section 1.2.2). For values of N within the current range of interest, the optimal choice of degree sum remains small (see [126, Section 3.1] for a relevant discussion). For example, the factorisation of a 768-bit RSA modulus by Kleinjung et al. [92] and the special number field sieve factorisation of $2^{1039} - 1$ by Aoki et al. [7] both used polynomial pairs which had a degree sum of 7. Corollary 2.1.4 shows that the restriction to small degree sum implies that a pair of number field sieve polynomials will necessarily have large coefficients.

A Note on Measuring Coefficient Size

The resultant of two coprime polynomials $f_1, f_2 \in \mathbb{Z}[x]$ is a homogeneous polynomial of degree $\deg f_1 + \deg f_2$ in their coefficients. As a result, some authors consider a pair of number field sieve polynomials $f_1, f_2 \in \mathbb{Z}[x]$ to have optimal coefficient size whenever $\mathrm{Res}(f_1, f_2) = \pm N$. In practical circumstances, the resultant does appear to provide a fair indication of coefficient size. However, the resultant only provides a lower bound on coefficient size. Therefore, on its own, the resultant of two polynomials may not serve as an accurate measure of coefficient size. This is demonstrated by the following lemma, which proves the existence of integer polynomial pairs with arbitrarily large coefficients and resultant equal to $\pm N$:

Lemma 2.1.6. Let X and s be positive real numbers. For integers $d \geq 2$ and $N > 1.5(d/\log 2)^d$, there exist degree d polynomials $f_1, f_2 \in \mathbb{Z}[x]$ such that $\mathrm{Res}(f_1, f_2) = \pm N$ and $\|f_i\|_{2,s} \geq X$, for $i = 1, 2$.

Proof. Let $m = \lfloor N^{1/d} \rfloor$ and write N in base-m:

$$N = a_d m^d + a_{d-1} m^{d-1} + \ldots + a_1 m + a_0,$$

where the coefficients $a_0, \ldots, a_d \in [0, m) \cap \mathbb{Z}$. Then $a_d = 1$, since $N > 1.5(d/\log 2)^d$ [49, Exercise 6.8]. Define polynomials

$$f_1(x) = c_1 \cdot (x - m) + \sum_{i=0}^{d} a_i x^i \quad \text{and} \quad f_2(x) = c_2 \cdot f_1(x) + (x - m),$$

where $c_1, c_2 \in \mathbb{Z}$, $c_2 \neq 0$, are chosen sufficiently large as to ensure $\|f_i\|_{2,s} \geq X$, for $i = 1, 2$. Then f_1 and f_2 are degree d integer polynomials and $f_1(m) = N$. Finally, by subtracting c_2 times row i of $\mathrm{Syl}(f_1, f_2)$ from row $d + i$, for $1 \leq i \leq d$, it follows that

$$\mathrm{Res}\,(f_1, f_2) = \mathrm{Res}\,(f_1(x), c_2 \cdot f_1(x) + (x - m)) = a_d^{d-1} \cdot \mathrm{Res}\,(f_1(x), x - m) = \pm f_1(m). \qquad \square$$

Remark 2.1.7. Lemma 2.1.6 additionally shows that the quotient $|\mathrm{Res}(f, g)| / \|f\|_{2,s}^{\deg g} \|g\|_{2,s}^{\deg f}$, for coprime polynomials $f, g \in \mathbb{R}[x]$, can be arbitrarily smaller than the upper bound of 1 provided by Lemma 2.1.3.

2.1.2 Quantifying Root Properties

The root properties of a number field sieve polynomial f can be quantified by the parameter $\alpha(f, y)$, introduced by Murphy [126] (denoted $\alpha(F)$ by Murphy). The parameter $\alpha(f, y)$ heuristically compares the effect of sieving on the polynomial values $F(a, b)$, for coprime pairs $(a, b) \in \mathbb{Z}^2$, with the effect of sieving on the integers. Thus $\alpha(f, y)$ provides a heuristic measure of the practical advantage or disadvantage resulting from a particular choice of polynomial.

The definition of $\alpha(f, y)$ is based on ideas developed for the analysis of the multiple polynomial quadratic sieve [22, 23] and the continued fractions method [94, Section 4.5.4]. Common to all three is the comparison of the behaviour of $\nu_p(z)$ as z ranges across different subsets of the integers. In particular, the parameter $\alpha(f, y)$ is obtained by comparing the behaviour for the integers with that of the polynomials values $F(a, b)$, for some polynomial $f \in \mathbb{Z}[x]$. To this end, for each prime p and irreducible $f \in \mathbb{Z}[x]$, the following quantities are introduced: define (if the limits exist)

$$\operatorname{cont}(p) = \lim_{X \to \infty} \frac{1}{X} \sum_{x \in [1,X] \cap \mathbb{Z}} \nu_p(x) \quad \text{and} \quad \operatorname{cont}(f,p) = \lim_{X \to \infty} \frac{1}{|\mathcal{A}(X)|} \sum_{(a,b) \in \mathcal{A}(X)} \nu_p(F(a,b)),$$

where $\mathcal{A}(X) = \{(a,b) \in \mathbb{Z}^2 \mid -X \leq a, b \leq X \text{ and } \gcd(a,b) = 1\}$, for all $X \geq 1$. In what follows, it is assumed that the limits always exist.

Consider the following idealised model for sieving a set $S \subseteq \mathbb{Z} \setminus \{0\}$ by all primes $p \leq y$:

1. an array is initialised with entries indexed by $z \in S$, such that entry z is equal to $\log |z|$; and

2. for each prime $p \leq y$ and each $z \in S$, $\nu_p(z) \cdot \log p$ is subtracted from the entry z.

If $S = \mathbb{Z} \setminus \{0\}$, then, in an informal sense, the value contained in entry $z \in S$ after sieving is, on average, equal to

$$\log |z| - \sum_{p \leq y} \operatorname{cont}(p) \cdot \log p, \tag{2.5}$$

where the sum ranges over all primes $p \leq y$. Similarly, if S contains polynomial values $F(a,b)$, for an irreducible polynomial $f \in \mathbb{Z}[x]$ and all coprime pairs $(a,b) \in \mathbb{Z}^2$, then the value contained in entry $F(a,b)$ after sieving is, on average, equal to

$$\log |F(a,b)| - \sum_{p \leq y} \operatorname{cont}(f,p) \cdot \log p.$$

For an irreducible polynomial $f \in \mathbb{Z}[x]$ and a real number $y > 0$, the parameter $\alpha(f,y)$ is defined by

$$\alpha(f,y) = \sum_{p \leq y} (\operatorname{cont}(p) - \operatorname{cont}(f,p)) \cdot \log p,$$

where the sum ranges over all primes $p \leq y$. Then

$$\log |F(a,b)| - \sum_{p \leq y} \operatorname{cont}(f,p) \cdot \log p = \log \left(|F(a,b)| \cdot e^{\alpha(f,y)} \right) - \sum_{p \leq y} \operatorname{cont}(p) \cdot \log p.$$

Comparing the right hand side with (2.5) suggests that a polynomial value $F(a,b)$, for a randomly chosen coprime pair $(a,b) \in \mathbb{Z}^2$, is, in an informal sense, as likely to be y-smooth as a randomly chosen integer of size $F(a,b) \cdot e^{\alpha(f,y)}$. This heuristic does not contradict the assumptions made in the complexity analysis of Section 1.2.2, as those were asymptotic considerations. However, it does suggest that in practice, it may be possible to produce polynomials with higher yield by leveraging root properties, i.e., by finding polynomials with negative α-values of large absolute value. This possibility was confirmed by the computational study of Murphy [126, Chapter 4] (see also [125]), where root properties were shown to exert a significant influence on polynomial yield. For example, Murphy [126, Section 4.2.4] used theoretical estimates of polynomial yield to show that varying α

across the "practical range" of α-values for the factorisation of the 140 digit number RSA-140 could influence yield by up to a factor of four. In particular, the example used a practical range of $[-7, 0]$, with the theoretical yield corresponding to an α-value of -7 found to be four times larger than the yield corresponding to an α-value of 0.

Estimation of $\alpha(f, y)$

In this section, estimates for $\mathrm{cont}(p)$ and $\mathrm{cont}(f, p)$ developed by Murphy and Montgomery [126, Section 3.2.2] are reviewed. These estimates allow for $\alpha(f, y)$ to be estimated in practice. In addition, the estimates provide information on how the distribution of a polynomials roots influences its α-value.

By definition, given a nonzero integer z, $\nu_p(z) = k$ if and only if $p^k \mid z$ and $p^{k+1} \nmid z$. Therefore,

$$\frac{1}{X} \sum_{x \in [1,X] \cap \mathbb{Z}} \nu_p(x) = \sum_{k=0}^{\left\lfloor \frac{\log X}{\log p} \right\rfloor} k \left[\frac{1}{p^k} \left(1 - \frac{1}{p}\right) + O\left(\frac{1}{X}\right) \right] = \sum_{k=0}^{\left\lfloor \frac{\log X}{\log p} \right\rfloor} \frac{k}{p^k} \left(1 - \frac{1}{p}\right) + O\left(\frac{\log^2 X}{X}\right),$$

for $X \to \infty$. It follows that

$$\mathrm{cont}(p) = \sum_{k=0}^{\infty} \frac{k}{p^k} \left(1 - \frac{1}{p}\right) = \frac{1}{p-1}.$$

In practice, $\mathrm{cont}(f, p)$ is usually estimated empirically: the approximation

$$\mathrm{cont}(f, p) \approx \frac{1}{|\mathcal{B}|} \sum_{(a,b) \in \mathcal{B}} \nu_p(F(a, b)) \tag{2.6}$$

is computed for a large set $\mathcal{B}$ of pairs (a, b) chosen uniformly at random from $\mathcal{A}(X)$, for some large value of X. However, it is illustrative to obtained a closed form expression for $\mathrm{cont}(f, p)$. Murphy [126, Section 3.2.2] used probabilistic arguments to obtain a (conjectured) closed form expression for $\mathrm{cont}(f, p)$, for all primes p such that $p \nmid \mathrm{disc}(f)$. Details absent from Murphy's arguments were then provided by Schmidt-Samoa [152, Section 4.2.2]. In the remainder of the section, the arguments of Murphy and Schmidt-Samoa are reviewed.

Given an irreducible polynomial $f \in \mathbb{Z}[x]$, if p is a prime such that the only root of $F(x, y)$ modulo p is the trivial root $(0, 0)$, then $\nu_p(F(a, b)) = 0$ for all coprime pairs $(a, b) \in \mathbb{Z}^2$, i.e., $\mathrm{cont}(f, p) = 0$. Therefore, for those primes p with $\mathrm{cont}(f, p) \neq 0$, it is necessary to consider the roots of the homogeneous polynomial $F(x, y)$ modulo p^k, for $k \geq 1$. To this end, define

$$V_{p^k} = \Big((\mathbb{Z}/p^k\mathbb{Z})^\times \times (\mathbb{Z}/p^k\mathbb{Z}) \Big) \cup \Big((\mathbb{Z}/p^k\mathbb{Z}) \times (\mathbb{Z}/p^k\mathbb{Z})^\times \Big), \quad \text{for all } k \geq 1.$$

Define an equivalence relation on V_{p^k} by $(x_1, y_1) \sim_{p^k} (x_2, y_2)$ if and only if $(x_1, y_1) = (\lambda x_2, \lambda y_2)$, for some $\lambda \in (\mathbb{Z}/p^k\mathbb{Z})^\times$. Then the projective line $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is defined to be the quotient $V_{p^k}/\sim_{p^k}$, and the equivalence class of $(x, y) \in V_{p^k}$ denoted $(x : y)$. For a prime p and $f \in \mathbb{Z}[x]$, a class

$(r_1 : r_2) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ such that $F(r_1, r_2) \equiv 0 \pmod{p^k}$ is called a *root* of $F(x, y)$ in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$.

The definition of the roots of $F(x, y)$ as classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ excludes from consideration the trivial roots $(0, 0)$ modulo p^k, for $k \geq 1$. Therefore, $\mathrm{cont}(f, p)$ can be estimated by determining the proportion of points $(a, b) \in \mathcal{A}(X)$ for which the class $(a : b) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is also a root of $F(x, y)$, for all $k \geq 1$ and $X \to \infty$. It follows from a result of Nymann [135, Theorem 3] that the pairs $(a + p^k\mathbb{Z}, b + p^k\mathbb{Z})$, for $(a, b) \in \mathcal{A}(X)$, are asymptotically equidistributed in V_{p^k}, for $k \geq 1$. For each $(x, y) \in V_{p^k}$, the corresponding class $(x : y) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ contains the elements $(\lambda x, \lambda y)$, for $\lambda \in (\mathbb{Z}/p^k\mathbb{Z})^\times$. Since either x or y is a unit in $\mathbb{Z}/p^k\mathbb{Z}$, it follows that the class $(x : y)$ contains exactly $|(\mathbb{Z}/p^k\mathbb{Z})^\times|$ elements of V_{p^k}. As a result, the classes $(a + p^k\mathbb{Z} : b + p^k\mathbb{Z})$, for $(a, b) \in \mathcal{A}(X)$, are asymptotically equidistributed in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$, for $k \geq 1$. Therefore, the limit as $X \to \infty$ of the proportion of points $(a, b) \in \mathcal{A}(X)$ for which the class $(a : b) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is also a root of $F(x, y)$ is equal to the proportion of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ that are roots of $F(x, y)$. The later proportion is determined for all but finitely many primes by the two lemmas that follow.

Lemma 2.1.8. For $k \geq 1$, the number of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is $p^{k-1}(p + 1)$.

Proof. Each class in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ contains exactly $|(\mathbb{Z}/p^k\mathbb{Z})^\times|$ elements of V_{p^k}. Therefore, the number of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is

$$\frac{|V_{p^k}|}{|(\mathbb{Z}/p^k\mathbb{Z})^\times|} = \frac{(p^k)^2 - (p^k - \varphi(p^k))^2}{\varphi(p^k)} = \frac{p^{2k-2}(p^2 - 1)}{p^{k-1}(p - 1)} = p^{k-1}(p + 1),$$

where φ is Euler's totient function. □

For each prime p and $f \in \mathbb{Z}[x]$, let $\sigma(f, p)$ denote the number of roots of $F(x, y)$ in $\mathbb{P}(\mathbb{Z}/p\mathbb{Z})$. Given a polynomial $f \in \mathbb{Z}[x]$, the following lemma determines, for all but finitely many primes p, the number of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ that are roots of $F(x, y)$, for $k \geq 1$:

Lemma 2.1.9. Let f be an integer polynomial and p be a prime such that $p \nmid \mathrm{disc}(f)$. Then $F(x, y)$ has exactly $\sigma(f, p)$ roots in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$, for all $k \geq 1$.

The proof presented here adapts arguments of Murphy [126, Section 3.2.2] and Schmidt-Samoa [152, Satz 4.27 and Satz 4.28].

Proof. Let $f = \sum_{i=0}^{d} a_i x^i$ be a degree d integer polynomial and p be a prime such that $p \nmid \mathrm{disc}(f)$. For $k \geq 1$, define $\overline{} : \mathbb{Z}/p^k\mathbb{Z} \to \mathbb{Z}/p\mathbb{Z}$ by $\overline{a + p^k\mathbb{Z}} = a + p\mathbb{Z}$. Then the induced map from $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ to $\mathbb{P}(\mathbb{Z}/p\mathbb{Z})$, defined by $(x : y) \mapsto (\overline{x} : \overline{y})$, sends roots of $F(x, y)$ in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ to roots in $\mathbb{P}(\mathbb{Z}/p\mathbb{Z})$. To complete the proof, it is shown that the induced map defines a bijection between the roots of $F(x, y)$ in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ and the roots in $\mathbb{P}(\mathbb{Z}/p\mathbb{Z})$, for all $k \geq 1$.

Suppose that $(r_1 : r_2) \in \mathbb{P}(\mathbb{Z}/p\mathbb{Z})$ is a root of $F(x, y)$ such that $r_2 \not\equiv 0 \pmod{p}$. Then $f(r_1 r_2^{-1}) \equiv 0 \pmod{p}$. Therefore, as p does not divide $\mathrm{disc}(f)$, Hensel's lemma implies that for each $k \geq 1$, there exists a unique element $x_k \in \mathbb{Z}/p^k\mathbb{Z}$ such that $f(x_k) \equiv 0 \pmod{p^k}$ and $\overline{x}_k \equiv r_1 r_2^{-1} \pmod{p}$. Hence, for each $k \geq 1$, $(x_k : 1) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is a root of $F(x, y)$ and

$$(\overline{x}_k, 1) \sim_p (\overline{x}_k r_2, r_2) \sim_p (r_1, r_2).$$

Moreover, if $(r_{1,k} : r_{2,k}) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is a root of $F(x, y)$ such that $(\overline{r}_{1,k}, \overline{r}_{2,k}) \sim_p (r_1, r_2)$, then the uniqueness of x_k implies that $r_{1,k} r_{2,k}^{-1} \equiv x_k \pmod{p^k}$. Thus,

$$(r_{1,k}, r_{2,k}) \sim_{p^k} (r_{1,k} r_{2,k}^{-1}, 1) \sim_{p^k} (x_k, 1).$$

Suppose now that $(r_1 : r_2) \in \mathbb{P}(\mathbb{Z}/p\mathbb{Z})$ is a root of $F(x, y)$ such that $r_2 \equiv 0 \pmod{p}$. Then 0 a root of $F(1, y)$ modulo p or, equivalently, p divides a_d. If f' denotes the derivative of f, then the discriminant of f and the resultant $\mathrm{Res}(f, f')$ are related as follows (see [99, p. 204]):

$$\mathrm{disc}(f) = (-1)^{\frac{d(d-1)}{2}} a_d^{-1} \mathrm{Res}(f, f').$$

Each entry in the first (resp. second) column of the Sylvester matrix $\mathrm{Syl}(f, f')$ is an integer multiple of a_d (resp. $\gcd(a_d, a_{d-1})$). It follows that $a_d \cdot \gcd(a_d, a_{d-1})$ divides $\mathrm{Res}(f, f')$. Therefore, p does not divide $\gcd(a_d, a_{d-1})$, since otherwise p divides $\mathrm{disc}(f)$. In particular, this implies that 0 is not a root of $F'(1, y)$ modulo p. Consequently, Hensel's lemma implies that for each $k \geq 1$, there exists a unique element $t_k \in \mathbb{Z}/p^k\mathbb{Z}$ such that $F(1, t_k) \equiv 0 \pmod{p^k}$ and $\bar{t}_k \equiv 0 \pmod{p}$. Hence, for each $k \geq 1$, $(1 : t_k) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is a root of $F(x, y)$ and

$$(1, \bar{t}_k) \sim_p (1, 0) \sim_p (r_1, 0) \sim_p (r_1, r_2).$$

Moreover, if $(r_{1,k} : r_{2,k}) \in \mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ is a root of $F(x, y)$ such that $(\overline{r}_{1,k}, \overline{r}_{2,k}) \sim_p (r_1, r_2)$, then the uniqueness of t_k implies that $r_{1,k}^{-1} r_{2,k} \equiv t_k \pmod{p^k}$. Thus,

$$(r_{1,k}, r_{2,k}) \sim_{p^k} (1, r_{1,k}^{-1} r_{2,k}) \sim_{p^k} (1, t_k). \qquad \square$$

Remark 2.1.10. The following terminology, introduced by Murphy [126, Section 3.2.2], is adopted throughout: for $f \in \mathbb{Z}[x]$, a root $(r_1 : r_2) \in \mathbb{P}(\mathbb{Z}/p\mathbb{Z})$ of the polynomial $F(x, y)$ is called *projective* if $r_2 \equiv 0 \pmod{p}$, and *non-projective* otherwise. Accordingly, a polynomial $f \in \mathbb{Z}[x]$ is said to have a projective root modulo p whenever p divides its leading coefficient. Similarly, an element $r \in \mathbb{Z}/p\mathbb{Z}$ such that $f(r) \equiv 0 \pmod{p}$ is referred to as a non-projective root of f modulo p.

Let $f \in \mathbb{Z}[x]$ and p be a prime such that $p \nmid \mathrm{disc}(f)$. Then Lemma 2.1.8 and Lemma 2.1.9 imply that the proportion of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ that are roots of $F(x, y)$ is

$$\frac{\sigma(f, p)}{p^{k-1}(p+1)}.$$

Hence, it is conjectured that

$$\operatorname{cont}(f,p) = \sum_{k=1}^{\infty} \frac{k\sigma(f,p)}{p^{k-1}(p+1)} \left(1 - \frac{1}{p}\right) = \sigma(f,p)\frac{p}{p^2-1}. \tag{2.7}$$

For primes that divide disc(f), computing the proportion of classes in $\mathbb{P}(\mathbb{Z}/p^k\mathbb{Z})$ that are roots of $F(x,y)$ is substantially more difficult (details are omitted here). For these primes, the approximation (2.6) may simply be used. Murphy [126, Section 3.2.2] performed computational experiments which compared (2.7) with the numerical approximation (2.6) for many polynomials. The two estimates were found to typically yield similar values.

From (2.7), it is clear that roots modulo small primes make the greatest contributions to a polynomial's α-value. Therefore, to obtain a negative α-value of large absolute value, it is important that a polynomial have many roots modulo small primes.

2.1.3 Ranking Polynomials

The most reliable method for determining those polynomial pairs with highest yield among a collection of pairs is to perform sieve experiments: an equal amount of sieving is performed for each pair and yield compared. However, for a large collection of pairs, performing sieve experiments may become too time-consuming. Therefore, efficient and reliable methods are needed for ranking polynomial pairs, according to their yield and without sieving. In this section, the now-standard method for ranking pairs in this manner, introduced by Murphy [126, Section 5.2], is reviewed.

Estimating Yield with the Dickman function

Recall that $\psi(x,y)$ denotes the number of y-smooth integers in the interval $[1,x]$. For nonnegative $u \in \mathbb{R}$, the *Dickman function* [51] is defined as follows:

$$\rho(u) = \lim_{x\to\infty} \frac{\psi(x, x^{1/u})}{x}, \text{ for } u > 1;$$

and $\rho(u) = 1$, otherwise. The Dickman function can be thought of as the limiting probability that a randomly chosen integer in $[1,x]$ is $x^{1/u}$-smooth. The function is known to satisfy

$$\psi(x, x^{1/u}) = x\rho(u) + \frac{x(1-\gamma)\rho(u-1)}{\log x} + O\left(\frac{x}{\log^2 x}\right), \tag{2.8}$$

where γ is Euler's constant [95]. Therefore, an estimate of the probability that an integer, chosen uniformly at random from $[1,x]$, is y-smooth is obtained from $\rho(u)$ by setting $u = \log x/\log y$. Moreover, (2.8) implies that this estimate is accurate to approximately $\log\log x/\log 10$ decimal places.

Given number field sieve polynomials f_1 and f_2, Kleinjung [91, Section 1] suggests the following "first approximation" of the number of relations found in the sieve stage:

$$\frac{6}{\pi^2} \sum_{(a,b)\in\mathcal{A}\cap\mathbb{Z}^2} \rho\left(\frac{\log|F_1(a,b)|}{\log y_1}\right)\rho\left(\frac{\log|F_2(a,b)|}{\log y_2}\right),$$

where y_i is the smoothness bound corresponding to f_i, for $i = 1, 2$; and the factor of $6/\pi^2$ accounts for the probability of $(a, b) \in \mathcal{A} \cap \mathbb{Z}^2$ satisfying $\gcd(a, b) = 1$ (see [9, p. 63]). This estimate neglects the effect of root properties on smoothness properties: recall from Section 2.1.2 that $F_i(a, b)$ is heuristically as likely to be y_i-smooth as a randomly chosen integer of size $F_i(a, b) \cdot e^{\alpha(f_i,y_i)}$. Accordingly, Kleinjung suggests that a better approximation of yield is provided by

$$\frac{6}{\pi^2} \sum_{(a,b)\in\mathcal{A}\cap\mathbb{Z}^2} \rho\left(\frac{\log|F_1(a,b)| + \alpha(f_1,y_1)}{\log y_1}\right)\rho\left(\frac{\log|F_2(a,b)| + \alpha(f_2,y_2)}{\log y_2}\right). \tag{2.9}$$

This estimate may be used to rank polynomials according to their yield. However, evaluating (2.9) is expensive in practice. Instead, it is preferable to use Murphy's method.

Murphy's $\mathbb{E}$-Value

Consider degree d number field sieve polynomials $f_1, f_2 \in \mathbb{Z}[x]$ that are to be sieved over respective regions $\mathcal{A}_1$ and $\mathcal{A}_2$ with equal area. In Section 2.1.1, the corresponding homogeneous polynomials F_1 and F_2 were shown to satisfy

$$\frac{\int_{\mathcal{A}_1} F_1(x,y)^2 dxdy}{\int_{\mathcal{A}_2} F_2(x,y)^2 dxdy} = \frac{\|f_1\|^2_{L^2,s_1}}{\|f_2\|^2_{L^2,s_2}},$$

where s_i denotes the skew of $\mathcal{A}_i$, for $i = 1, 2$. Therefore, given a number field sieve polynomial f with skew s, the size of F over the region $[-\sqrt{s}, \sqrt{s}\,] \times [\,0, 1/\sqrt{s}\,]$ provides sufficient information to allow f to be ranked according to its size properties. Murphy's method for ranking polynomials applies this observation to pairs of number field sieve polynomials by considering the sum (2.9) over points in $[-\sqrt{s}, \sqrt{s}\,]\times[\,0, 1/\sqrt{s}\,]$, rather than the integral points of the entire sieve region. The points considered by Murphy all lie on the ellipse

$$(x, y) = \left(\sqrt{s}\cos\theta, \frac{1}{\sqrt{s}}\sin\theta\right), \quad \text{for } \theta \in [0, 2\pi).$$

In particular, the points are given by k uniformly distributed values of $\theta \in [0, \pi]$:

$$\theta_i = \frac{\pi}{k}\left(i - \frac{1}{2}\right), \quad \text{for } i = 1, \ldots, k.$$

Given a pair of number field sieve polynomials $f_1, f_2 \in \mathbb{Z}[x]$ and the skew s of their sieve region, Murphy defines their rating $\mathbb{E}(f_1, f_2)$ (commonly referred to as Murphy's $\mathbb{E}$-value) as follows: let

$$u_i(\theta_j) = \frac{\log |F_i(\sqrt{s}\cos\theta_j, (1/\sqrt{s})\sin\theta_j)| + \alpha(f_i, y_i)}{\log y_i},$$

for $1 \leq i \leq 2$, $1 \leq j \leq k$; and

$$\mathbb{E}(f_1, f_2) = \sum_{j=1}^{k} \rho(u_1(\theta_j))\rho(u_2(\theta_j)). \tag{2.10}$$

Then the pairs of number field sieve polynomials in a collection are ranked according to descending order of $\mathbb{E}(f_1, f_2)$ values.

For the purpose of computing (2.10), Murphy [126, Section 2.2.1] suggests using the Patterson–Rumsey algorithm (described in the next section) to efficiently evaluate the Dickman function with small error. The reliability of $\mathbb{E}$ as a method for rating polynomials, according to yield and without sieving, has been examined by Murphy [126, Chapter 6]. There examples are provided which show that Murphy's $\mathbb{E}$-value consistently identifies polynomial pairs with highest actual yield.

Evaluating $\rho(u)$

In this section, the Patterson–Rumsey algorithm (as described by Bach and Peralta [10]) for evaluating the Dickman function is briefly described. The algorithm is based on the following observation made by Dickman [51]:

Theorem 2.1.11. *The Dickman function $\rho(u)$ is the (unique) continuous solution to the differential-difference equation*

$$u\rho'(u) + \rho(u-1) = 0, \quad \text{for } u > 1, \tag{2.11}$$

that satisfies the initial condition $\rho(u) = 1$, for $0 \leq u \leq 1$.

The differential-difference equation (2.11) implies that there exists an analytic function $\rho_k(u)$, for all integers $k \geq 1$, such that $\rho_k(u)$ agrees with $\rho(u)$ on the interval $[k-1, k]$. For example, it follows immediately from Theorem 2.1.11 that $\rho_1(u) = 1$ and $\rho_2(u) = 1 - \log u$. Bach and Peralta [10, Section 4] showed that the remaining functions, $\rho_k(u)$ for all integers $k \geq 3$, can be recursively computed by using the Taylor series of $\rho_{k-1}(u)$ to compute the Taylor series of $\rho_k(u)$. In particular, they proved the following:

Theorem 2.1.12. *Define real numbers $c_{k,i}$, for $k \geq 1$ and $i \geq 0$, as follows: $c_{1,0} = 1$, $c_{2,0} = 1 - \log 2$;*

$$\begin{aligned} c_{1,i} &= 0, \quad c_{2,i} = \frac{1}{i2^i}, && \textit{for } i \geq 1; \\ c_{k,i} &= \sum_{j=0}^{i-1} \frac{c_{k-1,j}}{ik^{i-j}}, && \textit{for } k > 2 \textit{ and } i > 0; \\ c_{k,0} &= \frac{1}{k-1} \sum_{j=1}^{\infty} \frac{c_{k,j}}{j+1}, && \textit{for } k > 2. \end{aligned}$$

Then for each positive integer k, the function $\rho_k(u) := \sum_{i=0}^{\infty} c_{k,i}(k-u)^i$ agrees with the Dickman function $\rho(u)$ on the interval $[k-1, k]$.

The Patterson–Rumsey algorithm precomputes the coefficients $c_{k,i}$, for all $(k,i) \in \{1, \ldots, k_{\max}\} \times \{0, \ldots, i_{\max}\}$, where $k_{\max}$ and $i_{\max}$ are parameters of the algorithm. Then given a nonnegative real $u_0 \leq k_{\max}$, the algorithm evaluates $\rho(u_0)$ by evaluating the truncated Taylor series of $\rho_{\lceil u \rceil}(u)$ at u_0. Bach and Peralta empirically found that computing the coefficients $c_{k,i}$ for $1 \leq i \leq 55$ is sufficient to give an relative error approximately equal to 10^{-17}, for all u in the range $0 \leq u \leq 20$. For numbers that are currently within reach of factorisation by the number field sieve, this range and error is adequate for the computation of Murphy's $\mathbb{E}$-Value.

2.2 Number Field Sieve Polynomial Generation

Current methods for polynomial selection employ a three-stage process. In the first stage, which herein is referred to as the *generation stage*, a large initial sample of polynomial pairs with good size and root properties is generated. In the second stage, the methods described in Section 2.1.3 are used to identify the best polynomial pairs in the sample without performing expensive sieve experiments. The third stage consists of using sieve experiments to identify, among those pairs that remain after the second stage, the polynomial pair with the highest yield in practice. In this section, existing methods for generating the initial sample of polynomial pairs are reviewed.

Existing algorithms for number field sieve polynomial generation are often divided into two classes. The first class, so-called *linear algorithms*, contains those generation algorithms that produce polynomial pairs such that one polynomial is linear. The second class, so-called *nonlinear algorithms*, contains those generation algorithms that produce pairs of nonlinear polynomials. The class of linear algorithms contains the base-m algorithm and its subsequent refinements by Montgomery and Murphy [126], and Kleinjung [91, 90]. The algorithms in this class have been used in a string of record factorisations [45, 33, 32, 11, 92] and remain the state of the art. The class of nonlinear algorithms contains Montgomery's two quadratics algorithm (see [54, Section 5] and [126, Section 2.3.1]) and its subsequent generalisations by Montgomery [119, 122], Williams [167], Prest and Zimmermann [149], and Koo, Jo and Kwon [97]. The algorithms in this class differ substantially from linear algorithms, with techniques from the geometry of numbers used to produce pairs of nonlinear polynomials with equal degree.

The linear algorithms of Montgomery and Murphy, and Kleinjung are reviewed in Section 2.2.1 and Section 2.2.2 respectively. An overview of the general approach used in nonlinear algorithms is provided in Section 2.2.3. An extensive review of nonlinear algorithms is delayed until Chapter 3. Finally, a lower bound on the performance of polynomial generation algorithms is derived in Section 2.2.4.

2.2.1 The Montgomery–Murphy Algorithm

In their algorithm, Montgomery and Murphy [126] introduced several improvements to the base-m method. The details of their improvements are reviewed here.

Overview of the Algorithm

The Montgomery–Murphy algorithm generates number field sieve polynomials in two stages: a modified base-m method is used to generated an initial polynomial pair, which is then either rejected or proceeds to the second stage where it is goes through an optimisation process. Throughout, the degree parameters d_1 and d_2 are fixed with $(d_1, d_2) = (d, 1)$, for some $d > 1$. Initial polynomial pairs are determined by parameters $(a_d, m) \in (\mathbb{Z} \setminus \{0\})^2$ such that $m \approx (N/a_d)^{1/d}$. For parameters (a_d, m), the initial polynomials

$$f_1 = a_d x^d + a_{d-1} x^{d-1} + \ldots + a_0 \quad \text{and} \quad f_2 = x - m,$$

are generated with f_1 obtained from the base-m representation of N by adding integer multiples of $x^i(x - m)$, $1 \le i \le d-1$, so that $|a_i| \le m/2$, for $0 \le i \le d-1$. Recall from Section 2.1.2 that the projective roots of a polynomial are determined by the prime factors of its leading coefficients. As a result, Montgomery and Murphy suggest choosing a_d to contain a product of small prime powers.

In the second stage, an initial polynomial pair (f_1, f_2) is optimised to find a new pair $(\tilde{f}_1, \tilde{f}_2)$ with improved size and root properties by two methods:

Translation: Set $\tilde{f}_1 = f_1(x - t)$ and $\tilde{f}_2 = f_2(x - t)$, for some $t \in \mathbb{Z}$. Then $\tilde{f}_1$ and $\tilde{f}_2$ have a common root $m + t$ modulo N.

Rotation: Set $\tilde{f}_1 = f_1 + r(x - m)$ and $\tilde{f}_2 = f_2$, for some $r \in \mathbb{Z}[x]$ with degree less than d.

The following lemma suggests that it is natural to consider rotation and translation as methods for improving the size and root properties of the initial polynomial pairs:

Lemma 2.2.1. Let f_1 be a nonzero degree d_1 integer polynomial such that $f_1(m_1/p_1)p_1^{d_1} = N$, for coprime integers m_1 and p_1 with $\gcd(p_1, N) = 1$. If $f_2 \in \mathbb{Z}[x]$ has degree $d_2 \le d_1$ and satisfies

$f_2(m_2/p_2) \equiv 0 \pmod N$, for integers m_2 and p_2 with $\gcd(p_2, N) = 1$, then there exist $t, q \in \mathbb{Z}$ and a polynomial $r \in \mathbb{Z}[x]$ of degree at most $d_1 - 1$ such that

$$f_2 = qf_1(x-t) + r(p_1(x-t) - m_1).$$

Proof. Choose an integer $t \equiv (p_1m_2 - m_1p_2)/p_1p_2 \pmod N$. Then there exists an integer k such that $f_2(m_1/p_1 + t)p_1^{d_2} = kN$. It follows that

$$f_2\left(\frac{m_1}{p_1} + t\right)p_1^{d_2} - kf_1\left(\frac{m_1}{p_1}\right)p_1^{d_1} = kN - kN = 0.$$

By assumption, $\gcd(m_1, p_1) = 1$ and $d_2 \leq d_1$. Therefore, Gauss' lemma implies that there exists a polynomial $r_0 \in \mathbb{Z}[x]$ of degree at most $d_1 - 1$ such that

$$f_2(x+t)p_1^{d_2} - kf_1(x)p_1^{d_1} = p_1^{d_2}r_0(x)(p_1x - m_1).$$

Hence,

$$f_2(x) = f_2((x-t)+t) = kf_1(x-t)p_1^{d_1-d_2} + r_0(x-t)(p_1(x-t) - m_1). \qquad \square$$

More generally, if $f_1(m_1/p_1)p_1^{d_1}$ is only known to equal some integer multiple of N, or $\gcd(m_1, p_1) \neq 1$, then only the following can be shown:

Lemma 2.2.2. Let $f_1, f_2 \in \mathbb{Z}[x]$ be nonzero of degree at most d such that there exist $m_i, p_i \in \mathbb{Z}$ with $f_i(m_i/p_i) \equiv 0 \pmod N$ and $\gcd(p_i, N) = 1$, for $i = 1, 2$. If $f_1(m_1/p_1) \neq 0$, then there exists an integer t, a rational number q, and a polynomial $r \in \mathbb{Q}[x]$ of degree at most $d - 1$ such that

$$f_2 = qf_1(x-t) + r(p_1(x-t) - m_1).$$

The proof of Lemma 2.2.2 applies similar ideas to that of Lemma 2.2.1 and is therefore omitted.

On one hand, translation does not alter root properties. Rather, it is used to improve size properties. On the other, rotation alters both size and root properties. The effect of rotations on root properties is exploited in the Montgomery–Murphy algorithm by the use of a sieve to identify rotations that provide good non-projective root properties. However, this process will often lead to an increase in the coefficient size of lower order terms. To compensate for this effect, and to allow rotations with large coefficients to be used (thus increasing supply and potential for finding good rotations), polynomials with large skew are sought. Such polynomials, often referred to as *highly skewed*, are also favourable for reason specific to the implementation of the sieve stage of the number field sieve (see [54, Section 6] and [90]).

The Montgomery–Murphy algorithm ensures that only highly skewed polynomials are found by allowing only those initial polynomials with a_d, a_{d-1} and a_{d-2} sufficiently small to continue through to the optimisation stage. Polynomials that survive are then subjected to two rotations phases. The first is aimed at producing a highly skewed polynomial with excellent size properties, while the second is aimed at producing a polynomial with excellent root properties. Those polynomials for which the first rotation phase does not produce a sufficiently small polynomial are rejected. In the second phase, rotations are restricted to low degrees to help retain size properties.

Throughout the algorithm, an initial rating is used to rejected polynomial pairs with poor size or root properties at several stages. The Murphy $\mathbb{E}$-value (see Section 2.1.3) is too expensive to compute for this task. Instead, a polynomial pair (f_1, f_2) is rated according to the size of

$$\inf_{s>0} \log \|f_1\|_{L,s} + \alpha(f_1, y), \tag{2.12}$$

where y is a bound on the primes considered in the sieve for the second phase of rotation. The rating ignores the properties of f_2 since each linear polynomial will have similar size and root properties. Moreover, $\log|F_2(a,b)|$ is relatively uniform over the sieve region when compared to $\log|F_1(a,b)|$. Therefore, those polynomial pairs for which (2.12) is smallest are likely to maximise (2.9) among all pairs found by the algorithm.

Summary of the Algorithm

To summarise the discussion of the previous section, the main steps of Montgomery-Murphy algorithm are now described. The description follows that given by Murphy [126, Procedure 5.1.6], which is implicitly optimised for degree 5 nonlinear polynomials, i.e., $d = 5$. Modifications to the description for $d \neq 5$ are briefly discussed in Remark 2.2.3.

For all integer pairs (a_d, m) such that $0 < a_d < a_{d,\max}$ contains a product of small prime powers, and $m \approx (N/a_d)^{1/d}$, the following steps are performed:

1. Compute the integral and fractional parts of

$$\frac{N - a_d m^d}{m^{d-1}} = a_{d-1} + \frac{a_{d-2}}{m} + O\left(\frac{1}{m^2}\right).$$

   If these two values are sufficiently small (i.e., a_{d-1} and a_{d-2} are sufficiently small), use the modified base-m method to compute an initial nonlinear polynomial f and proceed to Step 2. Otherwise, reject parameters (a_d, m) and proceed to the next pair.

2. Apply a multivariable optimisation procedure, such as the method described by Jedlička [81], to find real parameters t, c_0, c_1 and s that minimise the skewed L^2-norm (Definition 2.1.2) of the polynomial

$$f(x-t) + (c_1 x + c_0)((x-t) - m).$$

Compute the polynomial

$$\tilde{f} := f(x - \lceil t \rfloor) + (\lceil c_1 \rfloor x + \lceil c_0 \rfloor)((x - \lceil t \rfloor) - m).$$

If $\log \|\tilde{f}\|_{L^2,s}$ is sufficiently small, proceed to Step 3. Otherwise, proceed to the next pair of parameters (a_d, m).

3. For all $(j_0, j_1) \in \mathbb{Z}^2$, define

$$\tilde{f}_{j_1,j_0} = \tilde{f} + (j_1 x + j_0)((x - t) - m).$$

Use a sieve over primes $p \leq p_{\max}$ to find those integer pairs $(j_0, j_1) \in [-J_0, J_0] \times [-J_1, J_1]$, where $J_0 \ll J_1$, for which $\alpha(\tilde{f}_{j_1,j_0}, p_{\max})$ is sufficiently small. If no such pairs (j_0, j_1) are found, proceed to the next pair of parameters (a_d, m). Otherwise, proceed to Step 4 .

4. For each pair (j_1, j_0) found in the Step 3, compute an initial rating

$$\log \|\tilde{f}_{j_1,j_0}\|_{L^2,s} + \alpha(\tilde{f}_{j_1,j_0}, p_{\max}),$$

where s is from Step 2. For those pairs (j_1, j_0) with sufficiently small rating, perform a final translation of $\tilde{f}_{j_1,j_0}$ to improve size properties and compute the resulting polynomials skew. Add the polynomial, its corresponding linear polynomial, and its skew to the collection of polynomial pairs found. Proceed to the next pair of parameters (a_d, m).

Remark 2.2.3. Modifying the above description for $d \neq 5$ may require changing the degree of the rotations used in Step 2 and Step 3 and inspecting the size of more or less coefficients in Step 1. In order to not impinge upon the size properties of higher order coefficients, rotations of degree less than $(d - 1)/2$ should be used. As a result, for $d > 5$ it may be worthwhile to inspect the size of more higher order coefficients in Step 1.

2.2.2 Kleinjung's Algorithm

Kleinjung [91] suggested an improved method for generating the initial polynomials in the first step of the Montgomery–Murphy algorithm. The improvements made by Kleinjung are twofold. First, the linear polynomial is no longer required to be monic, leading to improved size and projective root properties. The second and more important improvement, is the introduction of an efficient method for determining parameters such that the coefficients a_d, a_{d-1} and a_{d-2} of the nonlinear polynomial $f_1 = \sum_{i=0}^{d} a_i x^i$ are small, without the explicit computation of a_{d-1} and a_{d-2}. Kleinjung's improvement of the Montgomery–Murphy algorithm remains the state of the art in number field sieve polynomial generation algorithms. In particular, the algorithm has been used in a string of record setting factorisations, culminating in that of RSA-768 [92]. The improvements made by Kleinjung are reviewed in this section.

Nonmonic Linear Polynomials

Kleinjung's algorithm replaces the modified base-m construction used in the Montgomery–Murphy algorithm with a construction based on finding base-(m,p) representations of N:

$$N = a_d m^d + a_{d-1} m^{d-1} p + \ldots + a_1 m p^{d-1} + a_0 p^d, \tag{2.13}$$

for integers $a_0, \ldots, a_d$, p and m. For each representation with $\gcd(m,p) = 1$ and $\gcd(p,N) = 1$, polynomials $f_1 = \sum_{i=0}^{d} a_i x^d$ and $f_2 = px - m$ are obtained with common root m/p modulo N and resultant $\operatorname{Res}(f_1, f_2) = \pm N$. Polynomials generated in this manner were first considered by Buhler et al. [29, Section 12.2]. For (2.13) to hold, it is necessary that the congruences

$$a_i m^i \equiv \frac{N - \sum_{j=i+1}^{d} a_j m^j p^{d-j}}{p^{d-i}} \pmod{p}, \quad \text{for } 0 \leq i \leq d, \tag{2.14}$$

are satisfied. Given consecutive coefficients $a_d, \ldots, a_k \in \mathbb{Z}$, such that $k \geq 1$ and (2.14) holds for $k \leq i \leq d$, a polynomial $f = \sum_{i=0}^{d} a_i x^d$, satisfying $f(m/p)p^d = N$, can be constructed by computing the remaining coefficients $a_{k-1}, \ldots, a_0$ with the following algorithm suggested by Kleinjung:

Algorithm 2.2.4.

Input: Coprime integers m and p. Integers $a_d, \ldots, a_k$ such that $k \geq 1$ and (2.14) holds for $k \leq i \leq d$.

Output: Integers $a_{k-1}, \ldots, a_0$ such that the polynomial $f = \sum_{i=0}^{d} a_i x^d$ satisfies $f(m/p)p^d = N$.

1. If $k = d$, set $r_k = N$; otherwise, set

$$r_k = \frac{N - \sum_{j=k+1}^{d} a_j m^j p^{d-j}}{p^{d-k}}.$$

2. For $i = k-1, \ldots, 0$, compute

$$r_i = \frac{r_{i+1} - a_{i+1} m^{i+1}}{p} \quad \text{and} \quad a_i = \frac{r_i}{m^i} + \delta_i,$$

where δ_i is chosen to satisfy $0 \leq \delta_i < p$ and $r_i \equiv a_i m^i \pmod{p}$.

3. Return $a_{k-1}, \ldots, a_0$.

By applying Algorithm 2.2.4 with $k = d$, Kleinjung [91, Lemma 2.1] obtained the following existence result:

Lemma 2.2.5. Let N, d, a_d, p and m be integers satisfying $a_d m^d \equiv N \pmod{p}$. Define $\tilde{m} = \sqrt[d]{\frac{N}{a_d}}$ and assume that $m \geq \tilde{m}$. Then there exists an integer polynomial $f = \sum_{i=0}^{d} a_i x^i$ such that

1. $f\left(\frac{m}{p}\right) \cdot p^d = N$;

2. $|a_{d-1}| < p + da_d\frac{m-\tilde{m}}{p}$; and

3. $|a_i| < p + m$, for $0 \leq i \leq d-2$.

Lemma 2.2.5 suggests extending the Montgomery–Murphy algorithm to search over positive integer parameters (a_d, m, p), where a_d and p both contain a product of small prime powers, $m \approx (N/a_d)^{1/d}$, and $a_d m^d \equiv N \pmod{p}$. Then Algorithm 2.2.4 may be used in Step 1 of the Montgomery–Murphy algorithm to compute the coefficients a_{d-1} and a_{d-2}. For those parameters (a_d, m, p) with $|a_{d-1}|$ and $|a_{d-2}|$ below respective bounds $a_{d-1,\max}$ and $a_{d-2,\max}$, the remaining steps of the algorithm can then be performed as normal. As noted by Kleinjung, the extended algorithm is slower than the original Montgomery–Murphy algorithm, since it is more time consuming to compute initial polynomials with Algorithm 2.2.4 instead of the modified base-m method. To address this problem, Kleinjung imposes restrictions on the now larger parameter space so that good parameters (a_d, m, p) can be efficiently identified without explicitly computing a_{d-1} and a_{d-2}. Details of this improvement are provided in the next section.

Identifying Good Parameters

Here the goal is efficiently identify parameters (a_d, m, p) for which there is a corresponding nonlinear polynomial $f = \sum_{i=1}^{d} a_i x^i$ such that $|a_d|$, $|a_{d-1}|$ and $|a_{d-2}|$ are below respective bounds $a_{d,\max}$, $a_{d-1,\max}$ and $a_{d-2,\max}$. Kleinjung [91, Section 3] discusses how to appropriately select these bounds. However, here they are treated as free parameters that satisfy $a_{d,\max} \leq a_{d-1,\max} \leq a_{d-2,\max}$. To begin, restrictions on the parameter space imposed Kleinjung are reviewed.

Kleinjung requires that a_d satisfies the inequalities $0 < a_d < a_{d,\max}$, with no further restrictions. Therefore, it is assumed that a_d is provided, leaving only the parameters p and m to be found. Given a_d and a prime $q \equiv 1 \pmod{d}$ such that $\gcd(q, a_d N) = 1$, the congruence $a_d x^d \equiv N \pmod{q}$ has either no solutions or d solutions. Kleinjung requires that the parameter p be a product of small distinct primes $p_1, \ldots, p_l$ such that $p_i \equiv 1 \pmod{d}$ and $\gcd(p_i, a_d N) = 1$, for $1 \leq i \leq l$. Given a parameter p of this form for which the congruence $a_d x^d \equiv N \pmod{p}$ has a solution, each of the d^l solutions are then indexed by a vector $\mu = (\mu_1, \ldots, \mu_l) \in \{1, \ldots, d\}^l$, and expressed in the form

$$x_\mu = \sum_{i=1}^{l} x_{i,\mu_i},$$

where $0 \leq x_{i,\mu_i} < p$ and $\frac{p}{p_i} \mid x_{i,\mu_i}$, for $1 \leq i \leq l$; and $x_{i,1}, \ldots, x_{i,d}$ are the d solutions of $a_d x^d \equiv N \pmod{p_i}$, for $1 \leq i \leq l$. Let m_0 be the least integer multiple of p greater than $\tilde{m} = (N/a_d)^{1/d}$. By setting $m_\mu = m_0 + x_\mu$, for each $\mu \in \{1, \ldots, d\}^l$, parameters (a_d, m_μ, p) satisfying $a_d m_\mu^d \equiv N \pmod{p}$ are obtained.

Lemma 2.2.6. With notation as above, let $a_{d-1} \in \mathbb{Z}$ satisfy

$$a_{d-1} m_\mu^{d-1} \equiv \frac{N - a_d m_\mu^d}{p} \pmod{p}, \tag{2.15}$$

and integers $a_0, \ldots, a_{d-2}$ be obtained by applying Algorithm 2.2.4 to inputs p, m_μ, a_d and a_{d-1}. If $a_{d-2}/m_\mu - \lfloor a_{d-2}/m_\mu \rceil < a_{d-2,\max}/m_\mu$, then there exists a polynomial $\tilde{f} = \sum_{i=0}^{d} \tilde{a}_i x^i$ such that

1. $\tilde{f}\left(\frac{m_\mu}{p}\right) \cdot p^d = N$;
2. $|\tilde{a}_{d-1}| < 2|a_{d-1}| + d a_d\,(l+1) + p\,(p/m_\mu + 1/2)$, $|\tilde{a}_{d-2}| < a_{d-2,\max}$; and
3. $|\tilde{a}_i| < p + m_\mu$, for $0 \le i \le d-3$.

Proof. Let $a_{d-1} \in \mathbb{Z}$ satisfy (2.15) and $a_0, \ldots, a_{d-2}$ be obtained by applying Algorithm 2.2.4 to p, m_μ, a_d and a_{d-1}. Then the polynomial $f = \sum_{i=0}^{d} a_i x^i$ satisfies $f(m_\mu/p)p^d = N$ and

$$|a_i| \le \frac{|r_i|}{m_\mu^i} + |\delta_i| \le \frac{|r_{i+1} - a_{i+1} m_\mu^i|}{p m_\mu^i} + p = \frac{|\delta_{i+1} m_\mu^{i+1}|}{p m_\mu^i} + p < p + m_\mu, \quad \text{for } 0 \le i \le d-3.$$

Let $\tilde{f} = f - \lfloor a_{d-2}/m_\mu \rceil \cdot (px - m_\mu)x^{d-2}$, and write $\tilde{f} = \sum_{i=0}^{d} \tilde{a}_i x^i$. Then $\tilde{f}(m_\mu/p)p^d = N$ and

$$\begin{aligned}
|\tilde{a}_{d-1}| &\le |a_{d-1}| + p\left(\frac{|a_{d-2}|}{m_\mu} + \frac{1}{2}\right) \le |a_{d-1}| + \frac{p}{m_\mu}\left(\frac{|r_{d-2}|}{m_\mu^{d-2}} + |\delta_{d-2}|\right) + \frac{p}{2} \\
&< |a_{d-1}| + \frac{|N - a_d m_\mu^d - a_{d-1} m_\mu^{d-1} p|}{m_\mu^{d-1} p} + \frac{p^2}{m_\mu} + \frac{p}{2} \\
&< |a_{d-1}| + \frac{d a_d (m_\mu - \tilde{m}) m_\mu^{d-1} + |a_{d-1}| m_\mu^{d-1} p}{m_\mu^{d-1} p} + p\left(\frac{p}{m_\mu} + \frac{1}{2}\right) \\
&< 2|a_{d-1}| + d a_d\,(l+1) + p\left(\frac{p}{m_\mu} + \frac{1}{2}\right).
\end{aligned}$$

Moreover, $|\tilde{a}_{d-2}| = |a_{d-1} - \lfloor a_{d-2}/m_\mu \rceil\, m_\mu| < a_{d-2,\max}$ and $|\tilde{a}_i| = |a_i|$, for $0 \le i \le d-3$. □

For $d = 5$ or $d = 6$, the bound on the coefficients $\tilde{a}_0, \ldots, \tilde{a}_{d-3}$ in Lemma 2.2.6 is sufficiently small. Therefore, if p is restricted to sufficiently small values (Kleinjung suggests that p should satisfy $p \le a_{d-1,\max}$ and $p \ll \tilde{m}$), then Lemma 2.2.6 implies that good parameters (a_d, m_μ, p) are found by identifying those $\mu \in \{1, \ldots, d\}^l$ such that there exists a small choice of a_{d-1} satisfying (2.15), and for which a_{d-2}/m_μ, where a_{d-2} is obtained from Algorithm 2.2.4, is within $a_{d-2,\max}/m_\mu$ of an integer.

For each $\mu \in \{1, \ldots, d\}^l$, define $a_{d-1,\mu}$ to be the coefficient a_{d-1} obtained from Algorithm 2.2.4 on the input of a_d, p and m_μ. Then all other permissible choices of the coefficient a_{d-1}, i.e., those that satisfy (2.15), are obtained as follows:

Lemma 2.2.7. With notation as above, $a_{d-1} \in \mathbb{Z}$ satisfies (2.15) if and only if there exist integers $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ such that $a_{d-1} = \sum_{i=1}^{l} e_{i,\mu_i}$ and

$$\begin{aligned} e_{1,\mu_1} &\equiv a_{d-1,(\mu_1,1,\ldots,1)} \pmod{p}, \\ e_{i,\mu_i} &\equiv a_{d-1,(\underbrace{1,\ldots,1}_{i-1},\mu_i,1,\ldots,1)} - a_{d-1,(1,\ldots,1)} \pmod{p}, \quad \text{for } 2 \leq i \leq l. \end{aligned} \tag{2.16}$$

Kleinjung [91, Lemma 3.4] proved the existence of integers $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ that satisfy (2.16) and for which $a_{d-1} = \sum_{i=1}^{l} e_{i,\mu_i}$ satisfies (2.15). Therefore, given an integer a'_{d-1} such that $a'_{d-1} = \sum_{i=1}^{l} e'_{i,\mu_i}$, for integers $e'_{1,\mu_1}, \ldots, e'_{l,\mu_l}$ that satisfy (2.16), then $a'_{d-1} \equiv a_{d-1} \pmod{p}$, and thus satisfies (2.15). Additionally, the converse of the lemma can be deduced from the existence result, since (2.15) determines a_{d-1} uniquely modulo p. Alternatively, Lemma 2.2.7 is proved directly as follows:

Proof. By construction, $a_{d-1,\mu}$ satisfies (2.15). As a result, and since $\gcd(m_\mu, p) = 1$, an integer a_{d-1} satisfies (2.15) if and only if $a_{d-1} \equiv a_{d-1,\mu} \pmod{p}$. Therefore, if there exist integers $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ such that $a_{d-1,\mu} = \sum_{i=1}^{l} e_{i,\mu_i}$ and (2.16) holds, then an integer a_{d-1} satisfies (2.15) if and only if (2.16) holds for integers

$$e'_{1,\mu_1} = e_{1,\mu_1} + (a_{d-1} - a_{d-1,\mu}) \quad \text{and} \quad e'_{i,\mu_i} = e_{i,\mu_i}, \text{ for } 2 \leq i \leq l,$$

which, by definition, satisfy $a_{d-1} = \sum_{i=1}^{l} e'_{i,\mu_i}$. Hence, it is sufficient to show that there exist integers $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ such that $a_{d-1,\mu} = \sum_{i=1}^{l} e_{i,\mu_i}$ and (2.16) holds.

Define l-dimensional vectors $\nu_i = (1, \ldots, 1, \mu_i, 1, \ldots, 1)$, where the entry μ_i appears in the ith coordinate, for $1 \leq i \leq l$. To prove the existence of integers $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ such that $a_{d-1,\mu} = \sum_{i=1}^{l} e_{i,\mu_i}$ and (2.16) holds, it is sufficient to show that

$$\Delta := a_{d-1,\mu} - a_{d-1,\nu_1} - \sum_{i=2}^{l} \left(a_{d-1,\nu_i} - a_{d-1,(1,\ldots,1)}\right) \equiv 0 \pmod{p}.$$

Multiplying (2.15) by $a_d m_\mu / N$ shows that

$$a_{d-1,\mu} \equiv \frac{a_d m_\mu}{N} \frac{N - a_d m_\mu^d}{p} \pmod{p}.$$

Analogous congruences hold for a_{d-1,ν_i}, for $1 \leq i \leq l$; and $a_{d-1,(1,\ldots,1)}$. Therefore, by substituting and rearranging, it follows that

$$\Delta \equiv \frac{a_d}{N}\left(m_\mu \frac{N - a_d m_\mu^d}{p} - m_{\nu_k} \frac{N - a_d m_{\nu_k}^d}{p} \right.$$

$$\left. - \sum_{i \neq k} \left(m_{\nu_i} \frac{N - a_d m_{\nu_i}^d}{p} - m_{(1,\ldots,1)} \frac{N - a_d m_{(1,\ldots,1)}^d}{p} \right) \right) \pmod{p},$$

for $1 \leq k \leq l$.

By construction, $m_\mu = m_0 + \sum_{i=1}^l x_{i,\mu_i}$, with similar expressions holding for $m_{\nu_1}, \ldots, m_{\nu_l}$ and $m_{(1,\ldots,1)}$. It follows that $m_\mu \equiv x_{k,\mu_k} \pmod{p_k}$, for $1 \leq k \leq l$, with similar congruences holding for $m_{\nu_1}, \ldots, m_{\nu_l}$ and $m_{(1,\ldots,1)}$. Moreover,

$$(m_\mu - m_{\nu_k}) - \sum_{i \neq k} \left(m_{\nu_i} - m_{(1,\ldots,1)} \right) = \left[(x_{k,\mu_k} - x_{k,\mu_k}) + \sum_{i \neq k} (x_{i,\mu_i} - x_{i,1}) \right] - \sum_{i \neq k} (x_{i,\mu_i} - x_{i,1}) = 0,$$

for $1 \leq k \leq l$. Therefore,

$$\begin{aligned} \Delta &\equiv -\frac{a_d^2}{\frac{p}{p_k} N} \left(\frac{m_\mu^{d+1} - m_{\nu_k}^{d+1}}{p_k} - \sum_{i \neq k} \frac{m_{\nu_i}^{d+1} - m_{(1,\ldots,1)}^{d+1}}{p_k} \right) \\ &\equiv -\frac{a_d}{\frac{p}{p_k}} \left(\frac{(m_\mu - m_{\nu_k})}{p_k} \frac{d a_d x_{k,\mu_k}^d}{N} - \sum_{i \neq k} \frac{\left(m_{\nu_i} - m_{(1,\ldots,1)} \right)}{p_k} \frac{d a_d x_{k,1}^d}{N} \right) \\ &\equiv -\frac{a_d d}{\frac{p}{p_k}} \frac{m_\mu - m_{\nu_k} - \sum_{i \neq k} \left(m_{\nu_i} - m_{(1,\ldots,1)} \right)}{p_k} \equiv 0 \pmod{p_k}, \end{aligned}$$

for $1 \leq k \leq l$. Hence, $\Delta \equiv 0 \pmod{p}$ as required. □

For each $\mu \in \{1, \ldots, d\}^l$, Kleinjung isolates the particular choice of the coefficient $a_{d-1} = \sum_{i=1}^l e_{i,\mu_i}$, where $e_{1,\mu_1}, \ldots, e_{l,\mu_l}$ are the unique integers that satisfy (2.16) and the inequalities $0 \leq e_{i,\mu_i} < p$, for $1 \leq i \leq l$. Then the prior restriction to small p guarantees that $|a_{d-1}|$ is small. To determine whether a_{d-2}/m_μ is near an integer, Kleinjung provides the approximation

$$\frac{a_{d-2}}{m_\mu} \approx \frac{N - a_d m_0^d}{p^2 m_0^{d-1}} - \sum_{i=1}^l \left(\frac{a_d d x_{i,\mu_i}}{p^2} + \frac{e_{i,\mu_i}}{p} \right),$$

where the error made is $O(dl^2(da_d + p)/m_0)$. Then, using a meet-in-the-middle algorithm, Kleinjung is able to identify in time $O(d^{l/2} \log d)$ those $\mu \in \{1, \ldots, d\}^l$ for which the approximation on the right hand side is sufficiently close to an integer. As a result, the average time spent checking one of the d^l choices of parameters (a_d, m_μ, p) is $O(d^{-l/2} \log d)$. Therefore, the parameter l should be chosen as large as possible.

The method of identifying good parameters described in this section has since been modified by Kleinjung [90], with the aim of producing polynomials with extremely large skews. Details of the modifications are not provided here.

2.2.3 Nonlinear Algorithms

Polynomials produced by linear algorithms experience an imbalance in the size of the values $F_1(a,b)$ and $F_2(a,b)$: for most pairs $(a,b) \in \mathbb{Z}^2$, the nonlinear polynomial produces values that are larger and thus less likely to be smooth. Nonlinear polynomial generation algorithms address this problem by producing pairs of nonlinear polynomials with equal or almost equal degrees. Current methods for nonlinear generation map the coefficients of number field sieve polynomials to vectors contained in some lattice. Then the problem of finding polynomials with small coefficients reduces to an instance of the well-studied problem of finding short vectors in a lattice. The principles behind the construction of those lattices used in current nonlinear algorithms are discussed in this section. To begin, the definition and some properties of lattices are introduced.

A *lattice* in $\mathbb{R}^n$ is a subgroup Λ of $\mathbb{R}^n$ with the following property: there exist $\mathbb{R}$-linear independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k \in \mathbb{R}^n$ such that $\Lambda = \sum_{i=1}^{k} \mathbb{Z}\boldsymbol{b}_i$. The vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ are said to form a *basis* for Λ, denoted throughout by a k-tuple $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$; and k is called the *dimension* or *rank* of Λ. When written with respect to the canonical orthonormal basis of $\mathbb{R}^n$, if $\boldsymbol{b}_i = (b_{i,1}, \ldots, b_{i,n})$, for $1 \le i \le k$, then the $k \times n$ matrix $B = (b_{i,j})_{1 \le i \le k, 1 \le j \le n}$ is called a *basis matrix* for Λ. Given a basis matrix B for Λ, the *determinant* of Λ is defined to be $\det \Lambda = \sqrt{\det BB^t}$. The determinant of a lattice is independent of the choice of basis. Algorithms for lattice reduction aim to produce bases consisting of short vectors. Given a basis for a k-dimensional lattice Λ, the LLL algorithm [103] returns a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ for Λ consisting of short vectors in polynomial time. In particular, for $k \ge 2$, the basis vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ satisfy the following inequalities:

$$\|\boldsymbol{b}_1\|_2 \le 2^{(k-1)/4} \det \Lambda^{\frac{1}{k}} \quad \text{and} \quad \|\boldsymbol{b}_1\|_2 \cdot \|\boldsymbol{b}_2\|_2 \le 2^{k-3}(k+4) \det \Lambda^{\frac{2}{k}}, \tag{2.17}$$

where $\|.\|_2$ is the Euclidean norm on $\mathbb{R}^n$.

Before nonlinear algorithms are discussed, it is illustrative to first review the lattice-based linear algorithm of Buhler et al. [29, Section 12.2]. Their algorithm is based on the observation that integer polynomials of bounded degree with a common root m modulo N can be characterised by an orthogonality condition on their coefficient vectors modulo N: an integer polynomial $f = \sum_{i=0}^{d} a_i x^i$ of degree *at most* d has m as a root modulo N if and only if the *coefficient vector* $(a_0, \ldots, a_d)$ is orthogonal to $(1, m, \ldots, m^d)$ modulo N. The set of all such coefficient vectors

$$L_{m,d} := \left\{ (a_0, \ldots, a_d) \in \mathbb{Z}^{d+1} \mid \sum_{i=0}^{d} a_i m^i \equiv 0 \bmod N \right\}, \tag{2.18}$$

forms a lattice in $\mathbb{Z}^{d+1}$ [29, Section 12.2]. A basis for $L_{m,d}$ is given by the $(d+1) \times (d+1)$ basis matrix

$$M_{m,d} = \begin{pmatrix} N & 0 & 0 & \dots & 0 \\ -m & 1 & 0 & \dots & 0 \\ -m^2 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -m^d & 0 & 0 & \dots & 1 \end{pmatrix}.$$

Therefore, $L_{m,d}$ is $(d+1)$-dimensional and $\det L_{m,d} = |\det M_{m,d}| = N$.

The algorithm of Buhler et al. begins with the selection of integers $m, p \approx N^{1/(d+1)}$ such that $\gcd(p, N) = 1$. Lattice reduction is then used to find a basis $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_{d+1})$ consisting of short vectors for the lattice $L_{t,d}$, for some integer $t \equiv mp^{-1} \pmod{N}$. At least one of the basis vectors $\boldsymbol{b}_i = (a_0, \dots, a_d)$ must satisfy $\sum_{i=0}^{d} a_i m^i p^{d-i} \neq 0$. For such a basis vector, the polynomials $f_1 = \sum_{i=0}^{d} a_i x^i$ and $f_2 = px - m$, with common root m/p modulo N, are taken. The bound on $\boldsymbol{b}_1$ in (2.17) suggests that the algorithm is capable of producing nonlinear polynomials with 2-norm of order $\det(L_{m,d})^{1/(d+1)} = N^{1/(d+1)}$. The algorithm is readily modified to produce skewed polynomials by introducing a skew parameter $s > 0$ and using lattice reduction to find a basis consisting of short vectors for the lattice with basis matrix $M_{t,d} \cdot \operatorname{diag}(S_0, \dots, S_d)$, where $S_i = s^{i-\frac{d}{2}}$, for $0 \leq i \leq d$. Once again, the skewed algorithm is capable of producing nonlinear polynomials f_1 with skewed 2-norm $\|f_1\|_{2,s}$ of order $N^{1/(d+1)}$, whenever $\|px - m\|_{2,s} \approx N^{1/(d+1)}$. However, both the skewed and non-skewed algorithms are not competitive with the algorithms of Montgomery and Murphy, and Kleinjung. As a result, lattice-based linear generation algorithms have received little attention in the literature.

Trivial modifications can be made to the algorithm of Buhler et al. in order to produce pairs of nonlinear polynomials with equal degree. However, the large determinant of $L_{m,d}$, and (2.17), imply that this approach is only expected to produce two degree d polynomials f_1 and f_2 such that $\|f_1\|_{2,s} \cdot \|f_2\|_{2,s}$ is of order $N^{2/(d+1)}$. When compared against the lower bound $\|f_1\|_{2,s} \cdot \|f_2\|_{2,s} \geq N^{1/d}$, provided by the resultant bound (Corollary 2.1.4), it follows that such a product of coefficient norms is likely far from optimal. To address this problem, current nonlinear algorithms employ a further modification of the approach, introduced by Montgomery (see [54, Section 5] and [126, Section 2.3.1]), whereby sublattices of $L_{m,d}$ with small determinant are constructed from "small" geometric progressions modulo N.

A *geometric progression* (GP) of *length* l and *ratio* r modulo N, denoted throughout by a vector $[c_0, \dots, c_{l-1}]$, is an integer sequence with the property that $c_i \equiv c_0 r^i \pmod{N}$, for $0 \leq i < l$. Central to the construction of lattices for nonlinear algorithms is the observation that

$$L_{m,d} = \left\{ (a_0, \dots, a_d) \in \mathbb{Z}^{d+1} \mid \sum_{i=0}^{d} a_i c_i \equiv 0 \bmod N \right\},$$

for any length $d+1$ geometric progression $[c_0, \dots, c_d]$, with ratio m modulo N, nonzero terms and

$\gcd(c_0, N) = 1$. Given such a geometric progression, nonlinear algorithms consider a sublattice of $L_{m,d}$ contained in the $\mathbb{Q}$-vector space orthogonal to $[c_0, \ldots, c_d]$. The role of N in the definition of the sublattice is therefore made implicit. Consequently, the determinant of the sublattice depends on the terms of the geometric progression, and not on N itself. Roughly speaking, a geometric progression with small terms is then expected to give rise to a sublattice of $L_{m,d}$ with small determinant. More generally, a lattice contained in the $\mathbb{Q}$-vector space orthogonal to multiple linearly independent geometric progressions is considered.

Montgomery's original algorithm (reported in [54, Section 5]) produces pairs of quadratic polynomials with optimal coefficient size, i.e., the resultant bound is attained (up to a constant factor). However, quadratic polynomial pairs are only competitive for the factorisation of integers containing at most 110-120 digits (see [126, Section 2.3.1]). To address this problem, Montgomery [119, 122] outlined a generalisation of the quadratic algorithm to arbitrary degrees. The generalisation relies on the construction of multiple geometric progressions from a long initial geometric progression. How to construct initial geometric progressions that meet the requirements of the generalisation remains a largely open problem.

Recent advances in geometric progression construction, in combination with relaxations of the requirements of Montgomery's approach, have lead to a string of new nonlinear algorithms. This line of research begins with Williams' [167] algorithms for producing quadratic and cubic polynomial pairs. Refinements to Williams' algorithms and extensions to arbitrary degree were provided by Prest and Zimmermann [149]. Finally, Koo, Jo and Kwon [97] extended methods for constructing geometric progressions. A detail review of these developments and Montgomery's algorithms appears in Chapter 3.

2.2.4 A Lower Bound on Polynomial Generation

Buhler et al. [29, Proposition 12.11] proved an asymptotic lower bound on the performance of any method for linear polynomial generation that is guaranteed to produce, for all N, polynomial pairs such that the nonlinear and linear polynomials have ∞-norms bounded by N^{t_1} and N^{t_2} respectively. In particular, they showed if such an algorithm produces nonlinear polynomials of degree at most d, then $N^{t_1+t_2}$ can not be expected to be substantially smaller than $N^{2/(d+2)}$. In this section, their result is extended to arbitrary methods of polynomial generation.

For positive integers d, C and a positive real s, let $\mathcal{M}(d, C, s)$ denote the set of non-constant polynomials $f \in \mathbb{Z}[x]$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C$. For positive integers d_1, d_1, C_1, C_2 and a positive real s, let $\mathcal{R}(d_1, d_2; C_1, C_2; s)$ denote the set of nonzero integers of the form $|\mathrm{Res}(f_1, f_2)|$, where $f_1 \in \mathcal{M}(d_1, C_1, s)$ and $f_2 \in \mathcal{M}(d_2, C_2, s)$. Then given an integer N such that there exists coprime polynomials $f_1 \in \mathcal{M}(d_1, C_1, s)$ and $f_2 \in \mathcal{M}(d_2, C_2, s)$ that share a common root modulo N, it follows that $\mathcal{R}(d_1, d_2; C_1, C_2; s)$ contains a nonzero multiple of N. Therefore, if there exists a polynomial generation algorithm with the property that, for each $N \in [1, X] \cap \mathbb{Z}$, the algorithm can produce coprime polynomials $f_1 \in \mathcal{M}(d_1, C_1, s)$ and $f_2 \in \mathcal{M}(d_2, C_2, s)$ with a common root modulo N, then it is necessary that $\mathcal{R}(d_1, d_2; C_1, C_2; s)$ contains a nonzero multiple of each integer in the interval $[1, X]$.

Theorem 2.2.8. *For each $\varepsilon > 0$, there exists a number $X(\varepsilon)$ with the following property. Suppose there exist positive integers d_1, d_2, C_1, C_2, X and a real number $s \geq 1$ such that*

1. *$X > X(\varepsilon)$;*
2. *$C_1 \geq s^{d_1/2}$, $C_2 \geq s^{d_2/2}$; and*
3. *$\mathcal{R}(d_1, d_2; C_1, C_2; s)$ contains a nonzero multiple of each integer in the interval $[1, X]$.*

Then $C_1 C_2 \geq 3^{-1} X^{\frac{2-\varepsilon}{d_1+d_2+1}}$.

Proof. Suppose positive integers d_1, d_2, C_1, C_2, X and a real number $s \geq 1$ satisfy the conditions of the theorem. Then Corollary 2.1.4 and the assumption that $\mathcal{R}(d_1, d_2; C_1, C_2; s)$ contains a nonzero multiple of X imply that

$$X \leq C_1^{d_2} C_2^{d_1}. \tag{2.19}$$

It may be assumed that $C_1 C_2 \leq X^{2/(d_1+d_2)}$, otherwise there is nothing left to prove. Then Lemma 2.1.3 implies that each element of $\mathcal{R}(d_1, d_2; C_1, C_2; s)$ has absolute value at most

$$C_1^{d_2} C_2^{d_1} \leq (C_1 C_2)^{d_1+d_2} \leq X^2.$$

Let $D = \max\{\tau(j) \mid j \in [1, X^2] \cap \mathbb{Z}\}$, where $\tau(j)$ denotes the number of divisors of j. Since each integer in the interval $[1, X]$ is required to have a nonzero multiple in $\mathcal{R}(d_1, d_2; C_1, C_2; s)$, it follows that

$$X \leq D \cdot |\mathcal{R}(d_1, d_2; C_1, C_2; s)|.$$

For all $f \in \mathbb{Z}[x]$, $\|f\|_{2,s} = \|-f\|_{2,s}$ and the inequality $\|f\|_{\infty,s} \leq \|f\|_{2,s}$ holds. Moreover, for all $f_1, f_2 \in \mathbb{Z}[x]$, the four resultants $\mathrm{Res}(\pm f_1, \pm f_2)$ have equal absolute value. Therefore,

$$|\mathcal{R}(d_1, d_2; C_1, C_2; s)| \leq \frac{1}{4} \prod_{i=1}^{2} |\mathcal{M}(d_i, C_i, s)| \leq \frac{1}{4} \prod_{i=1}^{2} \prod_{j=0}^{d_i} \left(2 C_i s^{\frac{d_i}{2} - j} + 1\right) \leq \frac{1}{4} \prod_{i=1}^{2} \prod_{j=0}^{d_i} 3 C_i s^{\frac{d_i}{2} - j},$$

where the final inequality follows from the inequalities $C_i \geq s^{d_i/2}$, for $i = 1, 2$. Consequently,

$$X \leq D \cdot |\mathcal{R}(d_1, d_2; C_1, C_2; s)| \leq 3^{d_1+d_2+1} D C_1^{d_1+1} C_2^{d_2+1}.$$

Multiplying this inequality by (2.19) gives

$$X^2 \leq 3^{d_1+d_2+1} D C_1^{d_1+d_2+1} C_2^{d_1+d_2+1} \tag{2.20}$$

The function τ satisfies $\tau(j) = j^{o(1)}$, for $j \to \infty$ (see [9, Theorem 13.12]). Therefore, given $\varepsilon > 0$, there exists a number $X(\varepsilon)$ such that $D \leq X^{\varepsilon}$, for all $X > X(\varepsilon)$. Hence, for all $X > X(\varepsilon)$, it follows from (2.20) that

$$C_1 C_2 \geq 3^{-1} D^{-\frac{1}{d_1+d_2+1}} X^{\frac{2}{d_1+d_2+1}} \geq 3^{-1} X^{\frac{2-\varepsilon}{d_1+d_2+1}}. \qquad \square$$

In the special case of Theorem 2.2.8 where $d_1 = d_2$, the resultant bound (Corollary 2.1.4) implies that C_1 and C_2 must satisfy the inequality $C_1 C_2 \geq X^{2/(d_1+d_2)}$, for all X.

Chapter 3

Nonlinear Polynomial Selection

Existing algorithms for number field sieve polynomial generation are divided into two classes: the class of linear algorithms, containing the algorithms of Montgomery–Murphy [126] and Kleinjung [91, 90]; and the class of nonlinear algorithms, containing those algorithms based on Montgomery's method. Linear algorithms have been employed in many record setting factorisations. However, polynomials produced by linear algorithms experience an imbalance in the size of the values $F_1(a, b)$ and $F_2(a, b)$ as a result of the large difference in their degrees. In contrast, nonlinear algorithms have received little practical attention, yet they produce pairs of nonlinear polynomials with equal or almost equal degrees. The lack of practical applications of nonlinear algorithm is explained by the fact that, until recently, nonlinear algorithms were only able to produce pairs of quadratic polynomials. Thus the range of numbers for which nonlinear algorithm were competitive with linear algorithms was restricted to integers of at most 110-120 digits (see [126, Section 2.3.1]). Consequently, their development has fallen behind that of linear algorithms.

Recent developments in nonlinear generation have broken the quadratic degree barrier, thus making nonlinear algorithms once again relevant to numbers in the current range of interest. In this chapter, these developments (see [167, 149, 97]) and Montgomery's algorithms (see [54, Section 5] and [119, 122]) are reviewed. Tools from the geometry of numbers are developed to aid in the analysis of nonlinear algorithms. In particular, they allow precise criteria for the selection of geometric progressions to be obtained. A family of geometric progressions modulo N, containing those used in existing algorithms, is characterised. The characterisation enables extensions to existing nonlinear algorithms to be made. Parameter selection for the extended algorithms is considered.

The importance of generating polynomials with a good combination of size and root properties was established in Chapter 2. However, existing nonlinear algorithms tend to focus solely on producing polynomials with small coefficients, thus leaving root properties to chance. As a result, rotations must be employed to improve the root properties of the polynomials they produce. In Chapter 4, a rotation free approach to generating polynomials with a good combination of size and root properties is developed. There methods developed in this chapter are incorporated into a much more general setting. This chapter therefore acts as a stepping stone in that direction.

The remainder of the chapter is organised as follows. In Section 3.1, preliminaries on lattices relevant to this chapter and the remainder of the thesis are provided. The brief outline of nonlinear polynomial generation provided in Section 2.2.3 is built upon in Section 3.2, with a detailed review of the method and existing algorithms provided. There the analysis of nonlinear algorithms is aided by results obtained in Section 3.2.1 on the properties of orthogonal lattices. Finally, new nonlinear generation algorithms are introduced and analysed in Section 3.3 and Section 3.4.

3.1 Preliminaries on Lattices

Throughout this thesis, results and algorithms from the geometry of numbers are extensively used. Here necessary background on lattices and lattice algorithms is reviewed. The reader is referred to [36, 115, 108, 134] for further background on the concepts discuss in this section.

A *lattice* in $\mathbb{R}^n$ is a subgroup Λ of $\mathbb{R}^n$ with the following property: there exist $\mathbb{R}$-linear independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k \in \mathbb{R}^n$ such that $\Lambda = \sum_{i=1}^{k} \mathbb{Z}\boldsymbol{b}_i$. The vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ are said to form a *basis* for Λ, denoted throughout by a k-tuple $\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$; and k is called the *dimension* or *rank* of Λ. If $k = n$, then Λ is referred to as *full-rank*. When written with respect to the canonical orthonormal basis of $\mathbb{R}^n$, if $\boldsymbol{b}_i = (b_{i,1}, \ldots, b_{i,n})$, for $1 \leq i \leq k$, then the $k \times n$ matrix $B = (b_{i,j})_{1 \leq i \leq k, 1 \leq j \leq n}$ is called a *basis matrix* for Λ. The *Gram matrix* of $\mathcal{B}$ is the $k \times k$ symmetric matrix BB^t. Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be bases for Λ with respective basis matrices B_1 and B_2. Then there exists a matrix $U \in \mathrm{GL}_k(\mathbb{Z})$ such that $UB_1 = B_2$. Thus the Gram matrix of $\mathcal{B}_2$ is $Q_2 = UQ_1U^t$, where Q_1 is the Gram matrix of $\mathcal{B}_1$. Therefore, the determinant of the Gram matrix is independent of the choice of basis. The *determinant* of Λ is defined to be $\det \Lambda = \sqrt{\det Q}$, where Q is the Gram matrix of one of its bases.

The *sublattices* of a lattice are its subgroups. A sublattice Λ' of a lattice Λ is referred to as a *full-rank sublattice* whenever $\dim \Lambda' = \dim \Lambda$. This occurs if and only if $[\Lambda : \Lambda']$ is finite. In this case, the determinants of Λ and Λ' satisfy the relationship $\det \Lambda' = [\Lambda : \Lambda'] \cdot \det \Lambda$. Let $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \mapsto \boldsymbol{x} \cdot \boldsymbol{y}$ denote the usual inner product in $\mathbb{R}^n$. The *dual lattice* of Λ is

$$\Lambda^{\times} = \{\boldsymbol{x} \in \mathrm{span}(\Lambda) \mid \langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}, \text{ for all } \boldsymbol{y} \in \Lambda\}.$$

For any basis $\mathcal{B}$ of Λ, the dual basis $\mathcal{B}^{\times}$ for $\mathrm{span}(\Lambda)$ is a basis for $\Lambda^{\times}$. A lattice with $\Lambda^{\times} = \Lambda$ is called *unimodular*. The lattice $\mathbb{Z}^n$ is unimodular.

Let $\|.\|_2$ be the norm on $\mathbb{R}^n$ induced by $\langle\ ,\ \rangle$. For a k-dimensional lattice Λ and all $1 \leq i \leq k$, the *ith minimum* $\lambda_i(\Lambda)$ of Λ is defined to be the minimum of $\max_{1 \leq j \leq i} \|\boldsymbol{v}_j\|_2$ over all sets of i linearly independent lattice vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_i\} \subset \Lambda$. Minkowski's second theorem (see [134, Theorem 5 p. 35]) provides an upper bound on the geometric mean of consecutive minima: if Λ is a k-dimensional lattice and t an integer satisfying $1 \leq t \leq k$, then

$$\left(\prod_{i=1}^{t} \lambda_i(\Lambda)\right)^{\frac{1}{t}} \leq \sqrt{\gamma_k}\, \det(\Lambda)^{\frac{1}{k}},$$

where γ_k denotes Hermite's constant (see [134, p. 20] for a definition). Hermite's constant is known [118, p. 17] to satisfy the linear bound $\gamma_k \leq 1 + k/4$, for all $k \geq 1$.

Algorithms for lattice reduction aim to produce bases consisting of short vectors. The most widely used reduction algorithm, due to Lenstra, Lenstra and Lovás [103], is the LLL algorithm. Given a basis for a lattice $\Lambda \subseteq \mathbb{Z}^n$, the LLL algorithm produces an LLL-reduced basis for Λ in polynomial time.

Definition 3.1.1. Let $\Lambda \subset \mathbb{R}^n$ be a k-dimensional lattice and $\mathcal{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ one of its bases. Let $(\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_k^*)$ be the Gram–Schmidt orthogonalisation of $\mathcal{B}$ and define $\mu_{i,j} = \langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle / \langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle$, for $1 \leq j < i \leq k$. Then $\mathcal{B}$ is *LLL-reduced* with factor $\delta \in (1/4, 1]$, if and only if the following conditions hold:

1. $|\mu_{i,j}| \leq 1/2$, for $1 \leq j < i \leq k$; and

2. $\left\|\boldsymbol{b}_{i+1}^* + \mu_{i+1,i}\boldsymbol{b}_i^*\right\|_2^2 \geq \delta \left\|\boldsymbol{b}_i^*\right\|_2^2$, for $1 \leq i < k$.

For simplicity, it is assumed throughout the thesis that LLL-reduced means LLL-reduced with factor $\delta = 3/4$. Accordingly, the following properties of LLL-reduced bases hold:

Theorem 3.1.2. *Let $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ be an LLL-reduced basis of a k-dimensional lattice $\Lambda \subset \mathbb{R}^n$. Then*

1. *$\|\boldsymbol{b}_1\|_2 \leq 2^{(k-1)/4} \det \Lambda^{1/k}$.*

2. *$\|\boldsymbol{b}_i\|_2 \leq 2^{(k-1)/2} \lambda_i(\Lambda)$, for $1 \leq i \leq k$.*

3. *If $\Lambda \subseteq \mathbb{Z}^n$, then $\|\boldsymbol{b}_i\|_2 \leq 2^{\frac{k(k-1)}{4(k-i+1)}} \det \Lambda^{\frac{1}{k-i+1}}$, for $1 \leq i \leq k$.*

The first two properties of the theorem were obtained by Lenstra et al. [103]. The third property is due to May [116, Theorem 4].

Given a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ of a k-dimensional lattice $\Lambda \subseteq \mathbb{Z}^n$, with $\max_i \|\boldsymbol{b}_i\|_2 \leq M$, the LLL algorithm returns an LLL-reduced basis in time $O(k^5 n \log^3 M)$ with arithmetic operations performed on integers of bit-length $O(k \log M)$. For instances where $\log M$ is large, it is preferable to use a floating point variant of the LLL algorithm such as the L^2 algorithm [132, 131]. The L^2 algorithm returns an LLL-reduced basis in time $O(k^4 n(k + \log M) \log M)$ and requires a precision of $(\log_2 3) \cdot k$ bits thus giving an improved overall complexity and requiring precision independent of $\log M$.

3.2 Nonlinear Polynomial Selection

There are two main problems that immediately arise from the approach to nonlinear polynomial generation described in Section 2.2.3: first, establishing a relationship between the size of terms in the geometric progressions and the determinant of the resulting lattices; and second, the construction of geometric progressions with small terms. In Section 3.2.1, tools are developed to address the first problem. There the object of study is the orthogonal lattice. A detailed description of nonlinear algorithms appears in Section 3.2.2. Based on the results of Section 3.2.1, criteria for the selection of geometric progressions are also provided. In Section 3.2.3, existing solutions to the second problem are reviewed.

Throughout the remainder of this chapter, big-O estimates may have implied constants depending on the degree parameter d.

3.2.1 The Orthogonal Lattice

Given a lattice $\Lambda \subseteq \mathbb{Z}^n$, denote by E_Λ the unique $\mathbb{Q}$-vector subspace of $\mathbb{Q}^n$ that is generated by any of its bases. The dimension of E_Λ over $\mathbb{Q}$ is equal to the dimension of Λ. Let $E_\Lambda^\perp$ denote the orthogonal complement of E_Λ with respect to $\langle\ ,\ \rangle$. For a lattice $\Lambda \subset \mathbb{Z}^n$ of dimension less than n, the *orthogonal lattice* of Λ is defined to be $\Lambda^\perp = \mathbb{Z}^n \cap E_\Lambda^\perp$. By a result due to Martinet [115, Proposition 1.3.4],

$$\dim \Lambda^\perp = \dim E_\Lambda^\perp = n - \dim \Lambda$$

if and only if the dimension of $(\mathbb{Z}^n)^\times \cap E_\Lambda^{\perp\perp}$ is equal to $\dim E_\Lambda$. The latter holds since $(\mathbb{Z}^n)^\times \cap E_\Lambda^{\perp\perp} = \mathbb{Z}^n \cap E_\Lambda$ is a lattice (see [115, Proposition 1.1.3]) which contains Λ as a sublattice.

Given a lattice $\Lambda \subseteq \mathbb{Z}^n$, let $\overline{\Lambda}$ denote the lattice $\mathbb{Z}^n \cap E_\Lambda$. Nguyen and Stern [130, Theorem 1] showed that for a lattice $\Lambda \subset \mathbb{Z}^n$ of dimension less than n, the determinants of Λ and $\Lambda^\perp$ are related as follows:

$$\det \Lambda = \left[\,\overline{\Lambda} : \Lambda\,\right] \cdot \det \Lambda^\perp.$$

Therefore, $\det \Lambda^\perp \leq \det \Lambda$, with equality if and only if $\Lambda = \overline{\Lambda}$. A lattice $\Lambda \subseteq \mathbb{Z}^n$ for which $\Lambda = \overline{\Lambda}$ holds is called *primitive*. Let B be a basis matrix for a k-dimensional lattice $\Lambda \subseteq \mathbb{Z}^n$. Then Λ is primitive if and only if the greatest common divisor of all $k \times k$ minors of B is 1 (see [153, Corollary 4.1c]). The following lemma determines the index $\left[\,\overline{\Lambda} : \Lambda\,\right]$ in general:

Lemma 3.2.1. Let $\Lambda \subseteq \mathbb{Z}^n$ be a k-dimensional lattice and B one of its basis matrices. Let Ω denote the greatest common divisor of all $k \times k$ minors of B. Then $\left[\,\overline{\Lambda} : \Lambda\,\right] = \Omega$.

Proof. Let $\overline{B}$ denote a basis matrix for $\overline{\Lambda}$. The lattice Λ is a full-rank sublattice of $\overline{\Lambda}$, thus there exists a $k \times k$ integer matrix U such that $B = U \cdot \overline{B}$ and $|\det U| = \left[\,\overline{\Lambda} : \Lambda\,\right]$. Hence, the lemma will follow by showing that $\Omega = |\det U|$.

For indices $1 \leq i_1 < \ldots < i_k \leq n$, let $B_{i_1,\ldots,i_k}$ (resp. $\overline{B}_{i_1,\ldots,i_k}$) denote the $k \times k$ submatrix of B (resp. $\overline{B}$) formed by columns $i_1, \ldots, i_k$. Then $B_{i_1,\ldots,i_k} = U \cdot \overline{B}_{i_1,\ldots,i_k}$, for all $1 \leq i_1 < \ldots < i_k \leq n$. Therefore, $\Omega = |\det U| \cdot \overline{\Omega}$, where $\overline{\Omega}$ is the greatest common divisor of all $k \times k$ minors of $\overline{B}$. However, $\overline{\Omega} = 1$ as the lattice $\overline{\Lambda}$ is primitive. □

The Determinant Under Transformation

For a k-dimensional lattice $\Lambda \subset \mathbb{R}^n$ and $S \in \mathrm{GL}_n(\mathbb{R})$, define $\Lambda_S = \{\boldsymbol{x} \cdot S \mid \boldsymbol{x} \in \Lambda\}$. Given a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ of Λ, define $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)_S = (\boldsymbol{b}_1 S, \ldots, \boldsymbol{b}_k S)$. Then Λ_S is a k-dimensional lattice in $\mathbb{R}^n$ with basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)_S$.

Lemma 3.2.2. Let $\Lambda \subset \mathbb{Z}^n$ be a lattice of dimension less than n, and $S \in \mathrm{GL}_n(\mathbb{R})$. Then

$$\det \Lambda_S^{\perp} = |\det S| \cdot \det \overline{\Lambda}_{S^{-t}},$$

where $S^{-t} = (S^{-1})^t$ denotes the inverse transpose of S.

Proof. Fix a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ for $\overline{\Lambda}$. The lattice $\overline{\Lambda}$ is primitive, thus $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)$ can be extended to a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ for $\mathbb{Z}^n$ [31, Lemma 2, Chapter 1]. Since $\mathbb{Z}^n$ is unimodular, the dual basis $(\boldsymbol{b}_1^{\times}, \ldots, \boldsymbol{b}_n^{\times})$ for $\mathbb{R}^n$ forms a basis for $\mathbb{Z}^n$. The dual basis is characterised by the equalities $\langle \boldsymbol{b}_i^{\times}, \boldsymbol{b}_j \rangle = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker delta. Therefore, $(\boldsymbol{b}_{k+1}^{\times}, \ldots, \boldsymbol{b}_n^{\times})$ forms a basis for the orthogonal lattice $\Lambda^{\perp}$. Hence, $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)_{S^{-t}}$ forms a basis for $\mathbb{Z}^n_{S^{-t}}$, $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)_{S^{-t}}$ forms a basis for $\overline{\Lambda}_{S^{-t}}$ and $(\boldsymbol{b}_{k+1}^{\times}, \ldots, \boldsymbol{b}_n^{\times})_S$ forms a basis for $\Lambda_S^{\perp}$.

For all $1 \leq i, j \leq n$,

$$\langle \boldsymbol{b}_i^{\times} S, \boldsymbol{b}_j S^{-t} \rangle = \boldsymbol{b}_i^{\times} S S^{-1} \boldsymbol{b}_j^{\,t} = \langle \boldsymbol{b}_i^{\times}, \boldsymbol{b}_j \rangle = \delta_{i,j}.$$

Thus $(\boldsymbol{b}_1^{\times}, \ldots, \boldsymbol{b}_n^{\times})_S$ is a dual basis of $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)_{S^{-t}}$. Therefore, by applying a result of Martinet [115, Corollary 1.3.5], with $E = \mathbb{R}^n$ and F equal to the subspace of $\mathbb{R}^n$ generated by $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k)_{S^{-t}}$, it follows that

$$|\det S|^{-1} = \det \mathbb{Z}^n_{S^{-t}} = \det\left(\overline{\Lambda}_{S^{-t}}\right) \cdot \det\left(\Lambda_S^{\perp}\right)^{-1}.$$ □

Given a basis for a lattice $\Lambda \subset \mathbb{Z}^n$, and a diagonal matrix $S \in \mathrm{GL}_n(\mathbb{R})$, the following theorem provides a method for computing the determinant of $\Lambda_S^{\perp}$:

Theorem 3.2.3. *Let $\Lambda \subset \mathbb{Z}^n$ be a lattice of dimension $k < n$, and B be one of its basis matrices. For all indices $1 \leq i_1 < \ldots < i_k \leq n$, denote by $B_{i_1,\ldots,i_k}$ the $k \times k$ submatrix of B formed by columns $i_1, \ldots, i_k$. For nonzero real numbers $S_1, \ldots, S_n$, define $S = \mathrm{diag}(S_1, \ldots, S_n)$. Then*

$$\det \Lambda_S^{\perp} = |S_1 \cdots S_n| \cdot \Omega^{-1} \cdot \sqrt{\sum_{1 \leq i_1 < \ldots < i_k \leq n} \left(\frac{\det B_{i_1,\ldots,i_k}}{S_{i_1} \cdots S_{i_k}} \right)^2},$$

where Ω is the greatest common divisor of all $k \times k$ minors of B.

Proof. The index of Λ in $\overline{\Lambda}$ is invariant under scaling by the matrix S^{-1}, i.e., $\left[\,\overline{\Lambda}_{S^{-1}} : \Lambda_{S^{-1}}\,\right] = \left[\,\overline{\Lambda} : \Lambda\,\right]$. Therefore, it follows from Lemma 3.2.1 and Lemma 3.2.2 that

$$\det \Lambda_S^{\perp} = |\det S| \cdot \det \overline{\Lambda}_{S^{-1}} = |S_1 \cdots S_n| \cdot \Omega^{-1} \cdot \det \Lambda_{S^{-1}}.$$

The matrix $P = BS^{-1}$ forms a basis matrix for $\Lambda_{S^{-1}}$. For all indices $1 \leq i_1 < \ldots < i_k \leq n$, let $P_{i_1,\ldots,i_k}$ denote the $k \times k$ submatrix of P formed by columns $i_1, \ldots, i_k$. Using the Cauchy-Binet formula (see [4, p. 86]) to compute $\det PP^t$, shows that

$$\det \Lambda_{S^{-1}} = \sqrt{\sum_{1 \leq i_1 < \ldots < i_k \leq n} \det \left(P_{i_1,\ldots,i_k}\right)^2},$$

The theorem then follows from the fact that $P_{i_1,\ldots,i_k} = B_{i_1,\ldots,i_k} \cdot \mathrm{diag}(S_{i_1}, \ldots, S_{i_k})^{-1}$, for all $1 \leq i_1 < \ldots < i_k \leq n$. □

Computing a Basis for the Orthogonal Lattice

Let Λ be a k-dimensional lattice in $\mathbb{Z}^n$ and $B = (b_{i,j})$ one of its basis matrices. A basis for the orthogonal lattice $\Lambda^{\perp}$ can be found by using either Algorithm 2.4.10 or Algorithm 2.7.2 described by Cohen [36] to compute a basis for the integer kernel of the matrix B. The former algorithm is based on Hermite normal form computation (see [36, Section 2.4.2]) and the latter algorithm on the MLLL algorithm of Pohst [140]. In practice, the MLLL based algorithm is preferable since it is more likely to avoid large integer arithmetic (see [36, Section 2.4.3]). Similarly, the LLL HNF algorithm of Havas, Majewski and Matthews [74, Section 6] may be used. If $M = \max_j \|(b_{1,j}, \ldots, b_{k,j})\|_2^2$, then the LLL HNF algorithm performs $O((n+k)^4 \log(nM))$ operation on integers of size $O(n \log(nM))$ [164]. The algorithm of Nguyen and Stern [130, Algorithm 5] directly computes an LLL-reduced basis for $\Lambda^{\perp}$. Given an $n \times n$ diagonal matrix S with integer entries and nonzero determinant, the following modification of their algorithm produces an LLL-reduced basis for $\Lambda_S^{\perp}$.

Algorithm 3.2.4.

Input: A basis matrix $B = (b_{i,j})_{1 \leq i \leq k, 1 \leq j \leq n}$ for a lattice $\Lambda \subset \mathbb{Z}^n$, where $k < n$. An $n \times n$ diagonal matrix S with integer entries and nonzero determinant.

Output: An LLL-reduced basis for $\Lambda_S^{\perp}$.

```
1. Select an integer $X > 2^{\frac{n-1}{2} + \frac{(n-k)(n-k-1)}{4}} \det \Lambda_S^{\perp}$.
```

2. Let $S = \operatorname{diag}(S_1, \dots, S_n)$. Compute the $n \times (n+k)$ matrix

$$D = \begin{pmatrix} S_1 & 0 & \dots & 0 & Xb_{1,1} & Xb_{2,1} & \dots & Xb_{k,1} \\ 0 & S_2 & \dots & 0 & Xb_{1,2} & Xb_{2,2} & \dots & Xb_{k,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_n & Xb_{1,n} & Xb_{2,n} & \dots & Xb_{k,n} \end{pmatrix}.$$

3. Compute an LLL-reduced basis $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ for the lattice with basis matrix D.

4. Let $\pi_\downarrow$ denote the projection that maps any vector in $\mathbb{R}^{n+k}$ to the vector in $\mathbb{R}^n$ obtained from its first n consecutive entries. Return the basis $\mathcal{X} = (\pi_\downarrow(\boldsymbol{x}_1), \dots, \pi_\downarrow(\boldsymbol{x}_{n-k}))$.

Accordingly, a generalisation of [130, Theorem 4] holds:

Theorem 3.2.5. *Algorithm 3.2.4 returns an LLL-reduced basis for $\Lambda_S^\perp$.*

Proof. Let Δ be the lattice with basis matrix D. Given a vector $\boldsymbol{y} = (y_1, \dots, y_n) \in \mathbb{Z}^n$,

$$\boldsymbol{y}D = (y_1 S_1, \dots, y_n S_n, X\langle \boldsymbol{y}, \boldsymbol{b}_1\rangle, \dots, X\langle \boldsymbol{y}, \boldsymbol{b}_k\rangle).$$

Therefore, $\boldsymbol{y} \in \Lambda^\perp$ if and only if $(y_1 S_1, \dots, y_n S_n, 0, \dots, 0) \in \Delta$. Consequently, if $\boldsymbol{x} \in \Delta$ and $\|\boldsymbol{x}\|_2 < X$, then $\pi_\downarrow(\boldsymbol{x}) \in \Lambda_S^\perp$.

The existence of an LLL-reduced basis for $\Lambda_S^\perp$ and Theorem 3.1.2, imply the existence of linearly independent vectors $\boldsymbol{y}_1, \dots, \boldsymbol{y}_{n-k} \in \Delta$ such that

$$\max_{1 \le i \le n-k} \|\boldsymbol{y}_i\|_2 \le 2^{\frac{(n-k)(n-k-1)}{4}} \det \Lambda_S^\perp.$$

Let $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$ be the LLL-reduced basis for Δ computed in Step 3 of the algorithm. Then Theorem 3.1.2 implies that

$$\max_{1 \le i \le n-k} \|\boldsymbol{x}_i\|_2 \le 2^{\frac{n-1}{2}} \max_{1 \le i \le n-k} \|\boldsymbol{y}_i\|_2 \le 2^{\frac{n-1}{2} + \frac{(n-k)(n-k-1)}{4}} \det \Lambda_S^\perp < X.$$

Thus $\mathcal{X} = (\pi_\downarrow(\boldsymbol{x}_1), \dots, \pi_\downarrow(\boldsymbol{x}_{n-k}))$ forms a basis for a sublattice of $\Lambda_S^\perp$. If $\mathcal{X}$ is not a basis of $\Lambda_S^\perp \subseteq \pi_\downarrow(\Delta)$, then there exist integers $z_{n-k+1}, \dots, z_n$, not all zero, such that the last k consecutive entries of the vector $\sum_{j=n-k+1}^{n} z_j \boldsymbol{x}_j$ are zero. That is, $\Lambda_S^\perp$ contains $n-k+1$ linearly independent vectors

$$\pi_\downarrow(\boldsymbol{x}_1), \dots, \pi_\downarrow(\boldsymbol{x}_{n-k}), \sum_{j=n-k+1}^{n} z_j \pi_\downarrow(\boldsymbol{x}_j),$$

which is absurd. Hence, $\mathcal{X}$ forms a basis for $\Lambda_S^\perp$.

It remains to show that $\mathcal{X}$ is LLL-reduced. From the definition of an LLL-reduced basis, it follows that $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-k})$ inherits the property of being LLL-reduced from $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$. If $(\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_{n-k}^*)$ is the Gram–Schmidt orthogonalisation of $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-k})$, then the last k consecutive entries of $\boldsymbol{x}_i^*$ must be 0, for $1 \leq i \leq n-k$. Therefore,

$$\left\langle \boldsymbol{x}_i, \boldsymbol{x}_j^* \right\rangle = \left\langle \pi_\downarrow(\boldsymbol{x}_i), \pi_\downarrow(\boldsymbol{x}_j^*) \right\rangle \quad \text{and} \quad \left\langle \boldsymbol{x}_i^*, \boldsymbol{x}_j^* \right\rangle = \left\langle \pi_\downarrow(\boldsymbol{x}_i^*), \pi_\downarrow(\boldsymbol{x}_j^*) \right\rangle, \quad \text{for } 1 \leq i, j \leq n-k.$$

Hence, the Gram–Schmidt orthogonalisation of $\mathcal{X}$ is equal to $(\pi_\downarrow(\boldsymbol{x}_1^*), \ldots, \pi_\downarrow(\boldsymbol{x}_{n-k}^*))$. Thus $\mathcal{X}$ inherits the property of being LLL-reduced from $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-k})$. □

As noted by Nguyen and Stern, the bounds on LLL-reduced bases (Theorem 3.1.2) are "quite pessimistic." Therefore, the lower bound on X occurring in Algorithm 3.2.4 can be reduced in practice. By using the L^2 algorithm in Step 3, Algorithm 3.2.4 takes time $O(n^4(n+k)(n+\log M)\log M)$, where M is an upper bound on the row vector norms of the matrix D from Step 2.

3.2.2 Nonlinear Polynomial Generation in Detail

To address the problem of constructing lattices with small determinants, the use of small geometric progressions modulo N was briefly introduced in Section 3.2. To make matters more concrete, the ideas introduced there are now discussed in detail.

Nonlinear algorithms search for polynomials with coefficient vectors contained in the lattice orthogonal to linearly independent geometric progressions with ratio m modulo N:

$$\boldsymbol{c}_1 = [c_{1,0}, \ldots, c_{1,d}], \boldsymbol{c}_2 = [c_{2,0}, \ldots, c_{2,d}], \ldots, \boldsymbol{c}_k = [c_{k,0}, \ldots, c_{k,d}], \quad 1 \leq k < d.$$

Let L denote the k-dimensional lattice with basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$. Geometric progressions that are also rational geometric progressions must be avoided. Otherwise, any nonlinear polynomial with coefficient vector in $L^\perp$ will be reducible. In general, $L^\perp$ may not be a sublattice of $L_{m,d}$, defined in (2.18). However, $L^\perp \subseteq L_{m,d}$ whenever at least one GP $\boldsymbol{c}_i$ has nonzero terms and $\gcd(c_{i,0}, N) = 1$. To obtained skewed polynomials, a skew parameter $s > 0$ is introduced and weights $S_i = s^{i-d/2}$ computed for $0 \leq i \leq d$. With $S = \operatorname{diag}(S_0, \ldots, S_d)$, lattice reduction is then used to find an LLL-reduced basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d-k+1})_S$, with $\boldsymbol{b}_i \in L^\perp$, for the lattice $L_S^\perp$. Finally, those polynomials with corresponding coefficient vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are returned.

In practice, the weights S_i can be replaced by arbitrary positive real values. However, defining $S_i = s^{i-d/2}$ ensures that the length of a vector $(a_0 S_0, \ldots, a_d S_d) \in L_S^\perp$ and the skewed 2-norm of the corresponding polynomial $f = \sum_{i=0}^d a_i x^i$ are related:

$$\|f\|_{2,s} = s^{\frac{d-\deg f}{2}} \cdot \|(a_0 S_0, \ldots, a_d S_d)\|_2 .$$

Therefore, if the vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ correspond to degree d polynomials $f_1, f_2 \in \mathbb{Z}[x]$ with nonzero resultant, then Corollary 2.1.4 and Theorem 3.1.2 imply that

$$N^{\frac{1}{d}} \leq \|f_1\|_{2,s} \cdot \|f_2\|_{2,s} \leq 2^{d-k} \cdot \gamma_{d-k+1} \cdot \det(L_S^{\perp})^{\frac{2}{d-k+1}}. \tag{3.1}$$

Consequently, when aiming to produce two polynomials of equal degree $d \geq 2$, the determinant of $L_S^{\perp}$ is of optimal size whenever $\det L_S^{\perp} = O(N^{(d-k+1)/2d})$.

The determinant of $L_S^{\perp}$ can be computed exactly using Theorem 3.2.3. However, this approach does not provide a clear intuition as to the relationship between the size of $\det L_S^{\perp}$ and the size of the geometric progressions $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$. For $s > 0$ and $(x_0, \ldots, x_n) \in \mathbb{R}^{n+1}$, define

$$\|(x_0, \ldots, x_n)\|_{2,s} = \sqrt{\sum_{i=0}^{n} \left| x_i s^{i-\frac{n}{2}} \right|^2}.$$

Then a more illustrative relationship between the size of $\det L_S^{\perp}$ and the size of the geometric progressions is provided by the following theorem:

Theorem 3.2.6. *For linearly independent geometric progressions*

$$\boldsymbol{c}_1 = [c_{1,0}, \ldots, c_{1,d}], \boldsymbol{c}_2 = [c_{2,0}, \ldots, c_{2,d}], \ldots, \boldsymbol{c}_k = [c_{k,0}, \ldots, c_{k,d}], \quad 1 \leq k < d,$$

with ratio m modulo N and $\gcd(c_{1,0}, N) = 1$*, let L denote the lattice with basis $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k)$. Then $L_S^{\perp}$ is $(d-k+1)$-dimensional and*

$$\det L_S^{\perp} \leq \frac{1}{N^{k-1}} \|\boldsymbol{c}_1\|_{2,s^{-1}} \cdots \|\boldsymbol{c}_k\|_{2,s^{-1}}.$$

Proof. Observe that $\boldsymbol{c}_i - (c_{i,0} c_{1,0}^{-1}) \boldsymbol{c}_1 \equiv \boldsymbol{0} \pmod{N}$, for $2 \leq i \leq k$. Thus N^{k-1} divides each $k \times k$ minor of the basis matrix $(c_{i,j})$ for L. Hence, Lemma 3.2.1 and Lemma 3.2.2 imply that $L_S^{\perp}$ is a $(d-k+1)$-dimensional lattice and

$$\det L_S^{\perp} \leq (S_0 \cdots S_d) \cdot \frac{1}{N^{k-1}} \cdot \det L_{S^{-1}} = \frac{1}{N^{k-1}} \cdot \det L_{S^{-1}}.$$

The proof is completed by using Hadamard's inequality (see [153, Section 1.3]) to bound $\det L_{S^{-1}}$. □

Theorem 3.2.6 provides a simple criterion for selecting geometric progressions: for a given skew $s > 0$, the best geometric progressions $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$ are precisely those for which $\|\boldsymbol{c}_i\|_{2,s^{-1}}$ are small.

The construction of small geometric progressions is, by a large margin, the most difficult part of nonlinear polynomial generation. One approach to this problem, introduced by Montgomery [119, 122] and later extended by Koo, Jo and Kwon [97, Section 3], suggests constructing an initial GP $\boldsymbol{c} = [c_0, \ldots, c_{l-1}]$ of length l, where $d < l < 2d$. Then $l - d$ geometric progressions of length $d + 1$ are obtained by taking successive terms:

$$\boldsymbol{c}_1 = [c_0, \ldots, c_d], \boldsymbol{c}_2 = [c_1, \ldots, c_{d+1}], \ldots, \boldsymbol{c}_{l-d} = [c_{l-d-1}, \ldots, c_{l-1}].$$

If the vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{l-d}$ do not form a basis for an $(l-d)$-dimensional sublattice of $L_{m,d}$, then $\boldsymbol{c}$ is rejected. For $s > 0$, the product of the norms $\|\boldsymbol{c}_i\|_{2,s^{-1}}$ is bounded in terms of the initial GP:

$$\prod_{i=1}^{l-d} \|\boldsymbol{c}_i\|_{2,s^{-1}} = \prod_{i=1}^{l-d} s^{\frac{l-d-1}{2}-(i-1)} \cdot \|\boldsymbol{c}_i\|_{2,s^{-1}} \leq \|\boldsymbol{c}\|_{2,s^{-1}}^{l-d}.$$

Therefore, to generate two degree d polynomials with optimal size, Theorem 3.2.6 and (3.1) suggests that the initial geometric progression $\boldsymbol{c}$ should satisfy

$$\|\boldsymbol{c}\|_{2,s^{-1}} = O\left(N^{\frac{(2d-1)(l-d)-(d-1)}{2d(l-d)}}\right). \tag{3.2}$$

For fixed d, the exponent of N in (3.2) is a strictly increasing function of l. Therefore, the weakest size requirements on $\boldsymbol{c}$ occur for $l = 2d-1$ (corresponding to Montgomery's algorithm). For this case, the orthogonal lattice is 2-dimensional and thus two linearly independent vectors of shortest length can be found in polynomial time by using Lagrange's algorithm (often called Gauss' algorithm, see [133] and references therein). For large N, the problem of efficiently constructing geometric progressions satisfying (3.2) remains open for all parameters $(d, l) \notin \{(2,3), (3,5)\}$.

Koo, Jo and Kwon observed that at least one degree d polynomial can be obtained for all $l/2 \leq d < l$. Therefore, distinct degree polynomial pairs can be obtained by varying the parameter d. This approach allows for nonlinear algorithms to be applied to N of any size.

3.2.3 Existing Algorithms

In this section, existing nonlinear generation algorithms are briefly reviewed. A uniform analysis of the algorithms that appear in this section is provided in Section 3.3 and Section 3.4. Therefore, attention is limited to the methods of GP and basis construction employed in each algorithm. Examples are provided for comparison between the algorithms.

Montgomery's Two Quadratics Algorithm

In Montgomery's two quadratics algorithm (see [54, Section 5] and [126, Section 2.3.1]), geometric progressions of length $d+1=3$ are constructed by first selecting an integer $p \geq 2$ (usually chosen to be prime) such that $\gcd(p, N) = 1$ and N is quadratic residue modulo p. Then one of the two possible values of $m \in \mathbb{Z}$ satisfying $m^2 \equiv N \pmod{p}$ and $|m - N^{1/2}| \leq p/2$ is chosen. Finally, the GP is taken to be $[c_0, c_1, c_2] = [p, m, (m^2-N)/p]$, with ratio mp^{-1} modulo N. For any integer $t \equiv c_2 c_1^{-1} \pmod{c_0}$, the matrix

$$\begin{pmatrix} c_1 & -c_0 & 0 \\ \frac{c_1 t - c_2}{c_0} & -t & 1 \end{pmatrix},$$

forms a basis matrix for the orthogonal lattice of $[c_0, c_1, c_2]\mathbb{Z}$.

For a given skew $s > 0$, choosing $p = O(s^{-1}\sqrt{N})$ guarantees that (3.2) holds. As a result, Montgomery's algorithm is capable of producing polynomials with optimal coefficient size. However, the restriction to quadratic polynomials means that the algorithm is not suitable for N containing more than 110–120 digits [126, Section 2.3.1]. Examples of polynomials generated with Montgomery's two quadratics algorithm can be found in [54, Section 10].

The Williams and Prest–Zimmermann Algorithms

Williams [167, Chapter 4] introduced another length 3 GP construction for producing pairs of quadratic polynomials. Roughly speaking, the new geometric progressions are obtained by setting $p = 1$ in Montgomery's construction. Williams also provided a length 4 GP construction for producing pairs of cubic polynomials. In both of Williams' algorithms, the skew parameter is restricted to $s = 1$. Prest and Zimmermann [149] extended Williams' algorithms to include skews $s \neq 1$, leading to a reduction in coefficient norms for the cubic algorithm. In addition, they generalised their algorithm to arbitrary degrees.

In the algorithms of Williams and Prest–Zimmermann, geometric progressions of length $d + 1$ are constructed by first selecting an integer m with $|m^d - N| = O(N^{1-1/d})$. Then the GP is taken to be

$$[c_0, \ldots, c_d] = [1, m, \ldots, m^{d-1}, m^d - N],$$

with ratio m modulo N. The matrix

$$\begin{pmatrix} -c_1 & 1 & 0 & \ldots & 0 \\ -c_2 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -c_d & 0 & 0 & \ldots & 1 \end{pmatrix},$$

forms a basis matrix for the orthogonal lattice of $[c_0, \ldots, c_d]\mathbb{Z}$.

Examples of polynomials found with the Williams and Prest–Zimmermann algorithms are found in [167, Chapter 5] and [149]. For comparison between the algorithms of this section, the following example is considered throughout:

Example 3.2.7. Let

$$\begin{aligned} N = \text{c91} =& 4567176039894108704358752160655628192034927306 \\ & 969828397739074346628988327155475222843793393. \end{aligned}$$

With $m = \lceil N^{1/3} \rceil = 16591382811472719807945870792 18$, Williams [167, Chapter 5] obtained the cubic polynomials:

$$\begin{aligned} f_1 = & \ 8962732699933084116x^3 \\ & - 20270774434332188756x^2 \\ & - 9743458171161776159x \\ & + 98228473793261830482 \end{aligned} \qquad \begin{aligned} f_2 = & \ 6252620090665427710 1x^3 \\ & - 14141384745569713065 8x^2 \\ & - 16127969563769626489 2x \\ & - 8860140805740788449 1 \end{aligned}$$

The product of coefficient norms $\|f_1\|_2 \cdot \|f_2\|_2$ is approximately $N^{0.445}$. The product $\|f_1\|_{2,s} \cdot \|f_2\|_{2,s}$ is minimised for $s_{\text{opt}} \approx 1.763$ with $\|f_1\|_{2,s_{\text{opt}}} \cdot \|f_2\|_{2,s_{\text{opt}}} \approx N^{0.443}$.

Applying Prest and Zimmermann's algorithm with $m = \lceil N^{1/3} \rceil$ and $s = 10^8$, the following pair of cubic polynomials is obtained:

$$\begin{aligned} g_1 &= 10363104x^3 - 23437957x^2 - 21147168576512214234486x - 109084939899748327411476171840, \\ g_2 &= 4776851x^3 - 10803677x^2 + 15035277150411604802155 5x - 100087822514431510434061442231. \end{aligned}$$

The product of coefficient norms $\|g_1\|_{2,10^8} \cdot \|g_2\|_{2,10^8}$ is approximately $N^{0.422}$. The product $\|g_1\|_{2,s} \cdot \|g_2\|_{2,s}$ is minimised for $s_{\text{opt}} \approx 45278023$ with $\|g_1\|_{2,s_{\text{opt}}} \cdot \|g_2\|_{2,s_{\text{opt}}} \approx N^{0.419}$. Consequently, the polynomials g_1 and g_2 have an optimised product of coefficient norms that is approximately 147 times smaller than that of f_1 and f_2.

The Koo–Jo–Kwon Algorithms

Koo, Jo and Kwon [97, Section 4.1] generalised Montgomery's GP construction to arbitrary degrees. They construct geometric progressions of length $d+1$ by first selecting positive integers $p = O((kN)^{1/d})$ and $k = O(1)$ such that $x^d \equiv kN \pmod{p}$ has a nonzero solution. An integer m satisfying $m^d \equiv kN \pmod{p}$ and $|m - \sqrt[d]{kN}| \leq p/2$ is chosen. Then the GP is taken to be

$$[c_0, \ldots, c_d] = \left[p^{d-1}, p^{d-2}m, \ldots, m^{d-1}, \frac{m^d - kN}{p} \right],$$

with ratio mp^{-1} modulo N. This construction is seen to reduce to Montgomery's construction for parameters $d = 2$, $k = 1$; and the constructions of Williams and Prest–Zimmerman for $p = k = 1$.

The Koo–Jo–Kwon and Prest–Zimmermann algorithms each produce polynomials which satisfy the same theoretical bounds on coefficient norms (see Section 3.3.1). However, for any given N, the additional parameters p and k allow for a wealth of new geometric progressions to be constructed. As a result, polynomials with significantly smaller coefficients may be found in practice.

Example 3.2.8. Let N = c91. Applying the Koo–Jo–Kwon algorithm with $s = 10^8$, $k = 1$, $p = 776112641898$ and $m = \lceil N^{1/3} \rceil + 5$, the following pair of cubic polynomials is obtained:

$$\begin{aligned} h_1 &= 124932x^3 - 276x^2 + 5900202319055646056 26x + 79893857071973416869543365671, \\ h_2 &= 156165x^3 - 345x^2 + 7375252900759839175 07x - 31491724894685122411171756 2717. \end{aligned}$$

The product of coefficient norms $\|h_1\|_{2,10^8} \cdot \|h_2\|_{2,10^8}$ is approximately $N^{0.383}$. The product $\|h_1\|_{2,s} \cdot \|h_2\|_{2,s}$ is minimised for $s_{\text{opt}} \approx 106759349$ with $\|h_1\|_{2,s_{\text{opt}}} \cdot \|h_2\|_{2,s_{\text{opt}}} \approx N^{0.383}$.

By extending their length $d+1$ GP construction, Koo, Jo and Kwon [97, Section 4.2] obtained a construction for length $d+2$ geometric progressions. The construction begins by selecting positive integers $p = \Theta((kN)^{1/d})$ and $k = O(1)$ such that $x^d \equiv kN \pmod{p^2}$ has a nonzero solution $m = \Theta(p)$. Then the GP is taken to be

$$[c_0, \ldots, c_{d+1}] = \left[p^{d-1}, p^{d-2}m, \ldots, m^{d-1}, \frac{m^d - kN}{p}, \frac{m(m^d - kN)}{p^2} \right],$$

with ratio mp^{-1} modulo N. Koo, Jo and Kwon do not analyse their algorithm for skews $s \neq 1$. This analysis is undertaken in Section 3.4, where it is shown that the algorithm improves upon previous algorithms for $d \geq 3$, with polynomials of optimal size produced when $d = 3$. However, the improvement is offset in part by the additional complexity of determining suitable parameters m, p and k.

3.3 Length $d+1$ Construction Revisited

Each of the length $d+1$ GP constructions discussed in Section 3.2.3 led to geometric progressions $[c_0, \ldots, c_d]$ for which $[c_0, \ldots, c_{d-1}]$ forms a rational GP. The following theorem determines all such geometric progressions that, in addition, satisfy the properties necessary for polynomial generation:

Theorem 3.3.1. *Let $[c_0, \ldots, c_d]$ be a GP modulo N with $d \geq 2$ and nonzero terms. Suppose that the following properties are satisfied:*

1. $\gcd(c_0, N) = 1$*;*

2. $[c_0, \ldots, c_{d-1}]$ *is a rational GP; and*

3. $[c_0, \ldots, c_{d-1}, c_d]$ *is not a rational GP.*

Then there exist nonzero integers a p, m and k, with $\gcd(m,p)=1$*, such that*

$$[c_0,\ldots,c_d]=\left[ap^{d-1},ap^{d-2}m,\ldots,am^{d-1},\frac{am^d-kN}{p}\right]. \tag{3.3}$$

Proof. Let $[c_0,\ldots,c_d]$ satisfy the conditions of the theorem. Then the second property implies the existence of nonzero integers a p and m, with $\gcd(m,p)=1$, such that $c_i=ap^{d-i-1}m^i$, for $0\le i\le d-1$. Consequently, $\gcd(ap,N)=1$ as a result of the first property, and

$$c_{\lfloor d/2\rfloor}c_{\lceil d/2\rceil}-c_0c_d=ap^{d-2}\left(am^d-pc_d\right).$$

Therefore, $c_{\lfloor d/2\rfloor}c_{\lceil d/2\rceil}-c_0c_d\neq 0$, since otherwise $[c_0,\ldots,c_d]$ forms rational GP, violating the third property. However, $[c_0,\ldots,c_d]$ is a GP modulo N, with ratio $r\equiv mp^{-1} \pmod N$. Thus

$$c_{\lfloor d/2\rfloor}c_{\lceil d/2\rceil}-c_0c_d\equiv\left(c_0r^{\lfloor d/2\rfloor}\right)\left(c_0r^{\lceil d/2\rceil}\right)-c_0\left(c_0r^d\right)\equiv c_0^2\left(r^{\lfloor d/2\rfloor+\lceil d/2\rceil}-r^d\right)\equiv 0 \pmod N.$$

Hence, $am^d-pc_d=kN$, for some nonzero $k\in\mathbb{Z}$. □

Given an arbitrary GP $[c_0,\ldots,c_{l-1}]$ with nonzero terms and length $l\ge 3$, $[c_0,c_1]$ forms a rational GP with ratio $c_1c_0^{-1}$. The following corollary is therefore a direct consequence of Theorem 3.3.1:

Corollary 3.3.2. Let $[c_0,\ldots,c_{l-1}]$ is a GP modulo N with $l\ge 3$ and nonzero terms. Suppose that the following properties are satisfied:

1. $\gcd(c_0,N)=1$; and
2. $[c_0,\ldots,c_{l-1}]$ is not a rational GP.

If $2\le d<l$ is the largest index such that $[c_0,\ldots,c_{d-1}]$ forms a rational GP, then there exist nonzero integers a p, m and k, with $\gcd(m,p)=1$, such that $[c_0,\ldots,c_d]$ is given by (3.3).

As a consequence of Theorem 3.3.1, the following nonlinear generation algorithm is obtained:

Algorithm 3.3.3.

Input: An integer $d\ge 2$. Positive integers a, p, m and k such that $\gcd(ap,N)=1$, $\gcd(m,p)=1$, and $(am^d-kN)/p$ is a nonzero integer. A positive integer s.

Output: A pair of integer polynomials f_1 and f_2 with common root mp^{-1} modulo N.

1. Compute $c_i=ap^{d-i-1}m^i$, for $0\le i\le d-1$; and $c_d=(am^d-kN)/p$.
2. Compute weights $S_i=s^{i-d/2}$, for $0\le i\le d$.
3. Let $L=[c_0,\ldots,c_d]\mathbb{Z}$ and $S=\mathrm{diag}(S_0,\ldots,S_d)$. Use Algorithm 3.2.4 to compute an LLL-reduced basis $(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_d)_S$ for the lattice $L_S^{\perp}$ (see Remark 3.3.4 below).

4. For $i = 1, 2$, write $\boldsymbol{b}_i = (a_{i,0}, \ldots, a_{i,d})$ and return the polynomial $f_i = \sum_{j=0}^{d} a_{i,j} x^j$.

The length $d+1$ GP construction in Step 1 of Algorithm 3.3.3 reduces to the construction of Montgomery's two quadratics algorithm for parameters $d = 2$, $a = k = 1$; the constructions of the Williams and Prest–Zimmerman algorithms for $a = p = k = 1$; and the construction of the Koo–Jo–Kwon algorithm for $a = 1$. In the next section, parameter selection for Algorithm 3.3.3 is considered.

Remark 3.3.4. In Step 3 of Algorithm 3.3.3, a reduced basis for $L_S^{\perp}$ may be found by first computing an LLL-reduced basis for $L_{S'}^{\perp}$, where $S' = \operatorname{diag}(1, s, \ldots, s^d)$. Given a reduced basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)_{S'}$ for $L_{S'}^{\perp}$, the definition of LLL-reduced bases (Definition 3.1.1) then implies that $(s^{-\frac{d}{2}}\boldsymbol{b}_1, \ldots, s^{-\frac{d}{2}}\boldsymbol{b}_d)_{S'}$ is also reduced. The latter is equal to $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)_S$, a basis for $L_S^{\perp}$. The restriction in the algorithm to $s \in \mathbb{Z}$ ensures that a reduced basis for $L_{S'}^{\perp}$ is able to be found with Algorithm 3.2.4.

3.3.1 Parameter Selection for Algorithm 3.3.3

Throughout this section, notation from Algorithm 3.3.3 is retained. In addition, let $\boldsymbol{c} = [c_0, \ldots, c_d]$. Then the polynomials f_1 and f_2 satisfy

$$f_i\left(\frac{m}{p}\right) \cdot p^d = \frac{p}{a}\langle \boldsymbol{b}_i, \boldsymbol{c}\rangle + \frac{a_{i,d}\,k}{a}N = \frac{a_{i,d}\,k}{a}N, \quad \text{for } i = 1, 2. \tag{3.4}$$

In Section 2.1.2, it was noted that root properties play a key role in determining the yield of number field sieve polynomials. There polynomial roots were divided into two classes: projective and non-projective. When $a = 1$, Koo, Jo and Kwon [97, Remark 5] noted that choosing k to contain a product of small primes improves the non-projective root properties of f_1 and f_2. More generally, (3.4) shows that selecting a and k to contain small prime factors may be used to aid both projective and non-projective root properties. However, the parameters a and k should be chosen small as $a/\gcd(a,k)$ divides $a_{i,d}$, for $i = 1, 2$; and $kN/\gcd(a,k)$ divides $\operatorname{Res}(f_1, f_2)$.

For $k = 1$, the parameter spaces of Algorithm 3.3.3 and Kleinjung's algorithm [91] coincide. Methods discussed by Kleinjung for efficiently generating parameters may be carried over to this setting and are readily extended to include $k \neq 1$. Additionally, parameter generation when $a = 1$ has been considered previously by Koo, Jo and Kwon [97, Section 4.1]. Consequently, the problem of generating parameters is not considered here. Instead, it is shown that under an appropriate choice of parameters, Algorithm 3.3.3 may be used to obtain degree d polynomials $f_1, f_2 \in \mathbb{Z}[x]$ with

$$\|f_i\|_{2,s} = O\left(N^{(1/d)(d^2-2d+2)/(d^2-d+2)}\right), \quad \text{for } i = 1, 2. \tag{3.5}$$

This yields polynomials of size $O(N^{1/4})$, for $d = 2$; $O(N^{5/24})$, for $d = 3$; and $O(N^{5/28})$, for $d = 4$. The exponent for $d = 2$ is optimal as a result of Corollary 2.1.4. The bound (3.5) is obtained without any assumptions on the size of vectors in LLL-reduced bases. This is in contrast to the previous analyses of [149, 97].

Applying Theorem 3.2.6, the determinant of $L_S^\perp$ satisfies

$$\det L_S^\perp \leq (S_0 \cdots S_d) \cdot \sqrt{\frac{c_0^2}{S_0^2} + \ldots + \frac{c_d^2}{S_d^2}} = \sqrt{\frac{c_0^2}{S_0^2} + \ldots + \frac{c_d^2}{S_d^2}}.$$

For $0 \leq i \leq d-1$,

$$\frac{c_i}{S_i} = ap^{d-i-1} m^i s^{\frac{d}{2}-i} = ap^{d-1} s^{\frac{d}{2}} \left(\frac{m}{ps}\right)^i.$$

Let $\tilde{m} = \sqrt[d]{\frac{kN}{a}}$ and assume $m \geq \tilde{m}$. Then

$$\frac{c_d}{S_d} = \frac{am^d - kN}{p} s^{-\frac{d}{2}} = \frac{a}{p}(m^d - \tilde{m}^d) s^{-\frac{d}{2}} < \frac{d(m-\tilde{m})}{ps} ap^{d-1} s^{\frac{d}{2}} \left(\frac{m}{ps}\right)^{d-1}.$$

Therefore, for parameters p and s satisfying $\sqrt{d}\,(m-\tilde{m}) \leq ps \leq m$,

$$\det L_S^\perp \leq \sqrt{2d}\, as^{1-\frac{d}{2}} m^{d-1}. \tag{3.6}$$

To minimise the determinant of $L_S^\perp$, it follows that the skew parameter s should be chosen as large as possible and $m \approx \tilde{m}$. However, the size of s is limited by the requirement that two degree d polynomials are found.

For a nonzero polynomial f with coefficient vector $\boldsymbol{x} \in L^\perp$ and degree less than d, (3.4) implies that $f(mp^{-1}) = 0$. Thus f must contain a monomial with nonzero coefficient divisible by m. Accordingly, the coefficient vector $\boldsymbol{x}$ satisfies $\|\boldsymbol{x}\|_{2,s} > s^{-d/2} m$. Therefore, if the basis vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ in the reduced basis for $L_S^\perp$ both satisfy $\|\boldsymbol{b}_i\|_{2,s} \leq s^{-d/2} m$, then f_1 and f_2 each have degree equal to d. Below it is shown that selecting s so that $\|\boldsymbol{b}_1\|_{2,s} \leq s^{-d/2} m$ holds is sufficient to guarantee that two degree d polynomial satisfying (3.5) can be found.

Theorem 3.1.2 and (3.6) imply that $\|\boldsymbol{b}_1\|_{2,s} \leq s^{-d/2} m$ whenever

$$2^{\frac{d-1}{4}} \left(\sqrt{2d}\, as^{1-\frac{d}{2}} m^{d-1}\right)^{\frac{1}{d}} \leq s^{-\frac{d}{2}} m.$$

Rearranging for s gives the bound

$$s \leq \frac{1}{\sqrt{2}} \left(\frac{m}{\sqrt{d}\, a}\right)^{\frac{2}{d^2-d+2}}.$$

Recall that s should be chosen as large as possible and $m \approx \tilde{m}$ in order to minimise the determinant of $L_S^\perp$. Therefore, parameters should be chosen to satisfy

$$m \geq \left(\frac{kN}{a}\right)^{\frac{1}{d}}, \quad s = \left\lfloor \frac{1}{\sqrt{2}} \left(\frac{m}{\sqrt{d}\, a}\right)^{\frac{2}{d^2-d+2}} \right\rfloor, \quad \sqrt{d}\,(m-\tilde{m}) \leq ps \leq m,$$

with $m = \Theta(\tilde{m})$. For such parameters, f_1 is of degree d with $f_1(mp^{-1}) \neq 0$, and substituting into the bound $\|\boldsymbol{b}_1\|_{2,s} \leq s^{-d/2}m$ shows that

$$\|f_1\|_{2,s} = O\left(a^{\frac{1}{d}\frac{2(d-1)}{d^2-d+2}} \cdot k^{\frac{1}{d}\frac{d^2-2d+2}{d^2-d+2}} \cdot N^{\frac{1}{d}\frac{d^2-2d+2}{d^2-d+2}}\right). \tag{3.7}$$

Setting $a = O(1)$ and $k = O(1)$ leads to f_1 satisfying the bound in (3.5).

Repeating the analysis for $m \leq \tilde{m}$ once again leads to parameters for which (3.7) is obtained. In both cases, the parameters satisfy

$$m = \Theta\left(\left(\frac{kN}{a}\right)^{\frac{1}{d}}\right), \quad s = \Theta\left(\left(\frac{kN}{a^{d+1}}\right)^{\frac{2}{d(d^2-d+2)}}\right), \quad \sqrt{d}\,|m - \tilde{m}| \leq ps \leq m. \tag{3.8}$$

For parameters satisfying (3.8), the condition $\|\boldsymbol{b}_1\|_{2,s} \leq s^{-d/2}m$ is now used to show that $\boldsymbol{b}_2$ satisfies $\|\boldsymbol{b}_2\|_{2,s} = O(s^{-d/2}m)$. Therefore, if the degree of f_2 is equal to d, then (3.5) holds. Otherwise, (3.5) is satisfied by the degree d polynomials f_1 and $f_1 + f_2$.

Assume (3.8) holds and $\|\boldsymbol{b}_1\|_{2,s} \leq s^{-d/2}m$. Then the vector $\boldsymbol{b} = (-m, p, 0, \ldots, 0)$ in $L^\perp$ satisfies

$$\|\boldsymbol{b}\|_{2,s} = \sqrt{(s^{-\frac{d}{2}}m)^2 + (s^{1-\frac{d}{2}}p)^2} \leq \sqrt{2}s^{-\frac{d}{2}}m.$$

Moreover, the vectors $\boldsymbol{b}_1, \boldsymbol{b} \in L^\perp$ are linearly independent since $\deg f_1 = d$. Hence, $\lambda_2(L_S^\perp) = O(s^{-d/2}m)$ and Theorem 3.1.2 implies that $\|\boldsymbol{b}_2\|_{2,s} = O(s^{-d/2}m)$.

Remark 3.3.5. The above arguments show that a degree d polynomial

$$f_{j_1,j_2,j_3}(x) = j_1 \cdot f_1(x) + j_2 \cdot f_2(x) + j_3 \cdot (px - m), \quad j_1, j_2, j_3 \in \mathbb{Z},$$

will satisfy $\|f_{j_1,j_2,j_3}\|_{2,s} = O(s^{-d/2}m)$ whenever $j_i = O(1)$, for $i = 1, 2, 3$. Therefore, it is possible to obtain multiple pairs of degree d polynomials that satisfy (3.5). Moreover, a sieve-like procedure such as that used in the Montgomery-Murphy algorithm [126, Procedure 5.1.6] (see also Section 2.2.1) may be used to identify polynomials f_{j_1,j_2,j_3} with good root properties.

3.4 The Koo–Jo–Kwon Length $d+2$ Construction Revisited

By utilising their length $d+2$ GP construction, Koo, Jo and Kwon [97, Corollary 4] obtained an algorithm for producing nonlinear polynomials of degree at most d such that the coefficient of x^{d-1} in each polynomial is equal to zero. Number field sieve polynomials with second highest coefficient equal to zero had previously been considered for linear algorithms by Kleinjung [90]. There the motivation was to produce polynomials with large skew in order to leverage practical advantages. In this section, it is shown that larger skews, when compared to those in Section 3.3.1, are able to be used in the Koo–Jo–Kwon algorithm. As a result, nonlinear polynomial pairs with smaller coefficient norms are obtained. To begin this section, minor improvements to the Koo–Jo–Kwon algorithm are now provided.

It follows immediately from Corollary 3.3.2 that the length $d+2$ GP construction of Koo, Jo and Kwon [97, Section 4.2] may be extended: if a, p, k, and m are positive integers that satisfy $\gcd(ap, N) = 1$, $\gcd(m, p) = 1$ and $am^d \equiv kN \pmod{p^2}$, then

$$[c_0, \ldots, c_{d+1}] = \left[ap^{d-1}, ap^{d-2}m, \ldots, am^{d-1}, \frac{am^d - kN}{p}, \frac{m(am^d - kN)}{p^2}\right], \tag{3.9}$$

is a GP with ratio mp^{-1} modulo N. The Koo–Jo–Kwon construction then corresponds to the special case where $a = 1$. Given a GP defined by (3.9), the proof of a result by Koo, Jo and Kwon [97, Corollary 4] is readily modified to show that an integer polynomial $f = \sum_{i=0}^{d} a_i x^i$ with coefficient vector orthogonal to both $[c_0, \ldots, c_d]$ and $[c_1, \ldots, c_{d+1}]$ must have $a_{d-1} = 0$. A stronger statement is provided by the following lemma:

Lemma 3.4.1. Let a, p, m, k and N be nonzero integers and $[c_0, \ldots, c_{d+1}]$ be defined by (3.9). For any vector $(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1}$, the following conditions are equivalent:

1. $(a_0, \ldots, a_d)$ is orthogonal to $[c_0, \ldots, c_d]$ and $[c_1, \ldots, c_{d+1}]$.
2. $a_{d-1} = 0$ and $(a_0, \ldots, a_d)$ is orthogonal to $(c_1, \ldots, c_{d-1}, 0, c_{d+1})$.

Proof. By construction,

$$[c_0, \ldots, c_d] - pm^{-1}[c_1, \ldots, c_{d+1}] = [0, \ldots, 0, m^{-1}kN, 0].$$

Hence, $(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1}$ is orthogonal to $[c_0, \ldots, c_d]$ and $[c_1, \ldots, c_{d+1}]$ if and only if $a_{d-1} = 0$ and $(a_0, \ldots, a_d)$ is orthogonal to vectors $(c_0, \ldots, c_{d-2}, 0, c_d)$ and $(c_1, \ldots, c_{d-1}, 0, c_{d+1})$, which are linearly dependent. □

Lemma 3.4.1 permits a somewhat smaller lattice to be used in the Koo–Jo–Kwon algorithm, thus offering a minor practical advantage. The improved algorithm may be described as follows:

Algorithm 3.4.2.

Input: An integer $d \geq 3$. Nonzero integers a, p, k and m such that $\gcd(ap, N) = 1$, $\gcd(m, p) = 1$, and $(am^d - kN)/p^2$ is a nonzero integer. A positive integer s.

Output: A pair of integer polynomials f_1 and f_2 with common root mp^{-1} modulo N.

1. Compute $c_i = ap^{d-i-2}m^i$, for $0 \leq i \leq d-2$; and $c_{d-1} = (am^d - kN)/p^2$.
2. Compute weights $S_i = s^{i-d/2}$, for $0 \leq i \leq d-2$; and $S_{d-1} = s^{d/2}$.

3. Let $L = (c_0, \ldots, c_{d-1})\mathbb{Z}$ and $S = \operatorname{diag}(S_0, \ldots, S_{d-1})$. By modifying the approach described in Remark 3.3.4, use Algorithm 3.2.4 to compute an LLL-reduced basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d-1})_S$ for $L_S^{\perp}$.

4. For $i = 1, 2$, write $\boldsymbol{b}_i = (a_{i,0}, \ldots, a_{i,d-2}, a_{i,d})$ and return the polynomial $f_i = a_{i,d}x^d + \sum_{j=0}^{d-2} a_{i,j}x^j$.

In the next section, parameter selection for Algorithm 3.4.2 is considered.

In Section 3.2.2, it was noted that a length l geometric progression may be used to generate degree d polynomials for all $l/2 \leq d < l$. Given a geometric progression $\boldsymbol{c} = [c_0, \ldots, c_{d+1}]$ defined by (3.9), it is therefore possible to generate polynomials of degrees d and $d+1$, for $d \geq 2$. Generating polynomials of degree less than d should not be considered as $[c_0, \ldots, c_{d-1}]$ forms a rational GP. A degree $d+1$ polynomial $f = \sum_{i=0}^{d+1} a_i x^i$ with coefficient vector orthogonal to $\boldsymbol{c}$ will satisfy

$$f\left(\frac{m}{p}\right) \cdot p^{d+1} = \frac{kN}{a}\left(a_{d+1}m + a_d p\right).$$

Hence, following the approach of Section 3.3.1 and choosing parameters so that $f(mp^{-1}) \neq 0$ is not sufficient to guarantee that f has degree equal to $d+1$. Parameter selection is therefore a more difficult problem, and is not addressed here.

3.4.1 Parameter Selection for Algorithm 3.4.2

Throughout this section, notation from Algorithm 3.4.2 is retained. In addition, let $\boldsymbol{c} = [c_0, \ldots, c_d]$. Then the polynomials f_1 and f_2 satisfy

$$f_i\left(\frac{m}{p}\right) \cdot p^d = \frac{p^2}{a}\langle \boldsymbol{b}_i, \boldsymbol{c}\rangle + \frac{a_{i,d}\,k}{a}N = \frac{a_{i,d}\,k}{a}N, \quad \text{for } i = 1, 2.$$

Therefore, similar to Section 3.3.1, the parameters a and k may be utilised to aid the root properties of f_1 and f_2. Generating parameters for Algorithm 3.4.2 is significantly more difficult than for Algorithm 3.3.3. This problem has, in effect, been considered by Kleinjung [90] and Koo–Jo–Kwon [97, Section 4.2]. Therefore, the problem is not considered here. Instead, the problem of selecting parameters that minimise the coefficient norms of f_1 and f_2 is considered.

Theorem 3.2.3 implies that

$$\det L_S^{\perp} \leq (S_0 \cdots S_{d-1}) \cdot \sqrt{\frac{c_0^2}{S_0^2} + \ldots + \frac{c_{d-1}^2}{S_{d-1}^2}} = s^{1-\frac{d}{2}} \cdot \sqrt{\frac{c_0^2}{S_0^2} + \ldots + \frac{c_{d-1}^2}{S_{d-1}^2}}.$$

By following the analysis of Section 3.3.1, the parameter space of Algorithm 3.4.2 can be restricted in such a way as to guarantee the degree of f_1 is equal to d and

$$\|f_1\|_{2,s} = O\left(a^{\frac{3d-4}{d(d^2-3d+4)}} \cdot p^{-\frac{d}{d^2-3d+4}} \cdot k^{\frac{1}{d}} \cdot N^{\frac{1}{d}}\right).$$

The restricted parameters then satisfy

$$m = \Theta\left(\left(\frac{kN}{a}\right)^{\frac{1}{d}}\right), \quad s = \Theta\left(\left(\frac{p}{a}\right)^{\frac{2}{d^2-3d+4}}\right), \quad \sqrt{d}\left|m - \left(\frac{kN}{a}\right)^{\frac{1}{d}}\right| \leq ps \leq m.$$

Clearly, the parameter p should be chosen as large as possible. By enforcing $p = \Theta(m/s)$,

$$\|f_1\|_{2,s} = O\left(a^{\frac{2(2d-3)}{d(d^2-3d+6)}} \cdot k^{\frac{d^2-4d+6}{d(d^2-3d+6)}} \cdot N^{\frac{d^2-4d+6}{d(d^2-3d+6)}}\right),$$

where $s = \Theta((kN/a^{d+1})^{(2/d)/(d^2-3d+6)})$. Similar to Section 3.3.1, if the degree of f_2 is not equal to d, then a second degree d polynomial can be found by considering linear combinations of the polynomials f_1, f_2 and $px - m$. Finally, by setting $a = O(1)$ and $k = O(1)$, it follows that Algorithm 3.4.2 can be used to obtain a pair of degree d polynomials with

$$\|f_i\|_{2,s} = O\left(N^{(1/d)(d^2-4d+6)/(d^2-3d+6)}\right), \quad \text{for } i = 1, 2.$$

This yields polynomials of size $O(N^{1/6})$, for $d = 3$; and $O(N^{3/20})$, for $d = 4$. The exponent for $d = 3$ is optimal as a result of Corollary 2.1.4.

Chapter 4

An Approach to Polynomial Selection

We emphasize that polynomial-searching is highly underdeveloped. There is much unexploited structure in the polynomial-searching problem. It appears far more tractable than factoring itself. Surely we can do better than brute force?

Bernstein and Lenstra [19]

Current best methods involve extensive searches, are guided by experience, helped by luck, and profit from patience.

Kleinjung et al. [92]

Current methods for polynomial selection generate polynomials in two stages: first polynomials with good size properties are found, then each polynomial is optimised by performing translation, rotation, and by computing its skew. There are two negative consequences that result from this approach. First, rotations are limited to low degree and small coefficient size in order to preserve the size properties of the initial polynomials. As a result, the expectation of finding good rotations is reduced, thus some luck is required to find polynomials with good root properties. This luck is usually provided in the form of a large initial sample of polynomials. However, this approach only amplifies the effect of the second negative consequence of the two-stage approach: only after optimisation can polynomials that rate poorly against measures such as the Murphy $\mathbb{E}$-value be discarded. Therefore, time is wasted on those polynomials that are generated, optimised, rated and then ultimately discarded.

In this chapter, a new approach to problem of number field sieve polynomial generation is developed. The approach targets only those polynomials with a good combination of size and root properties. As a result, the number of polynomials that require optimisation, and the time spent optimising each polynomial, are reduced. The development of the approach begins in Section 4.1, where new light is shed on the underlying algebraic structure of the polynomial generation problem by revisiting the resultant bound. At first, this may appear to be an unlikely place to start. However, the resultant bound was obtained by relating information about the combined size and root properties of two polynomials through upper and lower bounds on their resultant (see Section 2.1.1). By generalising the relationship hinted at by the proof of the resultant bound, it is shown that concepts from the theory of algebraic error-correcting codes apply naturally to the polynomial generation problem. This observation ultimately leads to the development of an approach to polynomial generation which is based on the framework for list decoding of algebraic error-correcting codes described by Guruswami et al. [69, Appendix A].

The approach of this chapter is not the first application of list decoding of algebraic error-correcting codes in number theory. Cheng and Wan [35] showed that a list decoding algorithm for Reed–Solomon codes can be used to find smooth polynomials over finite fields. This problem arises as part of the index calculus algorithm for computing discrete logarithms. Boneh [25] used a list decoding algorithm for Chinese remainder codes to find smooth integers in short intervals. Additionally, Boneh gave an algorithm for finding smooth values of a univariate polynomial. A final example is provided in Chapter 5, where a list decoding algorithm for a family of number field codes is used to find smooth algebraic integers in a number field.

The remainder of the chapter is organised as follows. Section 4.2 contains technical results on the divisibility properties of resultants, which allow for the resultant bound to be generalised. The generalised bound is then combined with a purely combinatorial result in Section 4.3, and used to derive bounds on the existence of number field sieve polynomials with small coefficients and good non-projective root properties. In Section 4.4, an initial realisation of the approach described in Section 4.1 is developed. The resulting algorithm is analysed and its performance compared against the combinatorially derived bounds of Section 4.3. To end the chapter, potential avenues for generalising the approach of Section 4.1 and improving its realisation are discussed in Section 4.5.

4.1 Overview of the Approach

The resultant bound was obtained by using information about the combined size and root properties of two coprime polynomials to provide respective upper and lower bounds on their resultant. In this section, the lower bound is generalised and used to establish a more general relationship between the size and root properties of two coprime polynomials. The relationship is then used to develop a new approach to polynomial generation. To begin, some notation that is used throughout the remainder of the chapter is introduced.

Define $\mathcal{U}$ to be the set of all integer pairs (p, r) such that p is prime and $0 \leq r < p$. Then the non-projective roots of a number field sieve polynomial f are in bijection with the pairs $(p, r) \in \mathcal{U}$ such that $f(r) \equiv 0 \pmod{p}$. For each element $(p, r) \in \mathcal{U}$, there is an associated maximal ideal $\mathfrak{p}_{p,r} := \langle p, x - r \rangle$ of $\mathbb{Z}[x]$. Given a nonzero polynomial $f \in \mathbb{Z}[x]$ and an ideal $\mathfrak{a} \subseteq \mathbb{Z}[x]$, define $\sigma(f, \mathfrak{a}) = 1$, if $f \in \mathfrak{a}$; and $\sigma(f, \mathfrak{a}) = 0$, otherwise. Furthermore, define $\sigma^*(f, \mathfrak{a})$ to be the maximum value $\sigma \geq 0$ such that $f \in \mathfrak{a}^\sigma$.

The lower bound used to obtain the resultant bound stems from the observation that the resultant of two coprime polynomials $f_1, f_2 \in \mathbb{Z}[x]$ with a common root modulo N belongs to the ideal $\langle f_1, f_2 \rangle \cap \mathbb{Z} \subseteq$

$\langle N\rangle$. Thus, their resultant must satisfy $N \leq |\mathrm{Res}(f_1, f_2)|$. A more general inequality can be obtained as an immediate consequence of the following lemma:

Lemma 4.1.1. Let N be a nonzero integer and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. If $f_1, f_2 \in \mathbb{Z}[x]$ are nonzero and f_1 is primitive, then

$$N^{\sigma^*(f_1, \langle N, x-m\rangle)\sigma^*(f_2, \langle N, x-m\rangle)} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f_1, \mathfrak{p}_i)\sigma^*(f_2, \mathfrak{p}_i)} \text{ divides } \mathrm{Res}(f_1, f_2),$$

for all $m \in \mathbb{Z}$.

A proof of Lemma 4.1.1 is provided in Section 4.2. Importantly, the lemma allows greater information about the combined root properties of two polynomials to be incorporated into the lower bound used to obtain the resultant bound. The tightened bound is summarised by the following lemma, which is the main technical result that underlies the approach of this chapter:

Lemma 4.1.2. Let N be a nonzero integer and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. Suppose there exist non-constant polynomials $f_1, f_2 \in \mathbb{Z}[x]$ such that f_1 is primitive and

$$\|f_1\|_{2,s}^{\deg f_2} \cdot \|f_2\|_{2,s}^{\deg f_1} < N^{\sigma^*(f_1, \langle N, x-m\rangle)\sigma^*(f_2, \langle N, x-m\rangle)} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f_1, \mathfrak{p}_i)\sigma^*(f_2, \mathfrak{p}_i)}, \tag{4.1}$$

for some $m \in \mathbb{Z}$ and $s > 0$. Then $\mathrm{Res}(f_1, f_2) = 0$.

Proof. Let $f_1, f_2 \in \mathbb{Z}[x]$ satisfy the conditions of the lemma and define

$$R(f_1, f_2) = N^{\sigma^*(f_1, \langle N, x-m\rangle)\sigma^*(f_2, \langle N, x-m\rangle)} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f_1, \mathfrak{p}_i)\sigma^*(f_2, \mathfrak{p}_i)}.$$

Then, on one hand, Lemma 4.1.1 implies that $R(f_1, f_2)$ divides $\mathrm{Res}(f_1, f_2)$. On the other hand, Lemma 2.1.3 and (4.1) imply that

$$|\mathrm{Res}(f_1, f_2)| \leq \|f_1\|_{2,s}^{\deg f_2} \cdot \|f_2\|_{2,s}^{\deg f_1} < R(f_1, f_2).$$

Hence, if $\mathrm{Res}(f_1, f_2) \neq 0$, then

$$R(f_1, f_2) \leq \|f_1\|_{2,s}^{\deg f_2} \cdot \|f_2\|_{2,s}^{\deg f_1} < R(f_1, f_2),$$

which is absurd. Therefore, $\mathrm{Res}(f_1, f_2) = 0$. □

The proof of Lemma 4.1.2 extends arguments made by Shparlinski and Steinfeld [154, Theorem 1]. The lemma may viewed as a generalisation of Howgrave-Graham's [78, Section 2] well-known sufficient condition for a modular root of an integer polynomial to also be an integer root.

Corollary 4.1.3. Let non-constant polynomials $f_1, f_2 \in \mathbb{Z}[x]$ be irreducible with $\deg f_i \leq d$ and $\|f_i\|_{2,s} \leq M$, for $i = 1, 2$. If

$$\sum_{(p,r)\in\mathcal{U}} \sigma^*(f_1, \mathfrak{p}_{p,r})\sigma^*(f_2, \mathfrak{p}_{p,r}) \log p > 2d \log M, \tag{4.2}$$

then $f_1 = \pm f_2$.

Proof. It follows from (2.3) that the resultant of two integer polynomials vanishes if and only if they have a nontrivial common divisor in $\mathbb{Q}[x]$. Therefore, suppose that $f_1, f_2 \in \mathbb{Z}[x]$ satisfy the conditions of the corollary and (4.2) holds. Then Lemma 4.1.2 implies that $\operatorname{Res}(f_1, f_2) = 0$, i.e., f_1 and f_2 have a nontrivial common divisor in $\mathbb{Q}[x]$. However, f_1 and f_2 are both primitive and irreducible over $\mathbb{Q}$. Hence, $f_1 = \pm f_2$. □

Corollary 4.1.3 shows that an irreducible polynomial $f \in \mathbb{Z}[x]$ is uniquely determined (up to units), given sufficient information about its non-projective roots. Here, "sufficient" depends on the coefficient size and the degree of f. Moreover, once sufficient information about the non-projective roots of f is known, any additional information may be viewed as redundant. This observation motivates the approach of this chapter: use ideas from the theory of algebraic error-correcting codes to exploit any redundant information. To realise this approach, ideas are adapted from the ideal-theoretic framework for list decoding of algebraic error-correcting codes described by Guruswami et al. [69, Appendix A] (see also [68, Section 7]). Explicitly, based on the framework and Lemma 4.1.2, the following approach to generating polynomials with root m modulo N and a good combination of size and root properties is deduced:

1. Choose pairwise comaximal ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i,r_i}$, with $p_i \nmid N$; and positive integer weights $z_0, \ldots, z_n$.
2. Find a nonzero polynomial $h \in \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ such that $\deg h$ and $\|h\|_{2,s}$ are small.
3. Factor h over $\mathbb{Q}$ and search among its factors for polynomials with root m modulo N and a good combination of size and root properties.

Given such a polynomial h, Lemma 4.1.2 guarantees that any non-constant irreducible polynomial $f \in \mathbb{Z}[x]$ with $f(m) \equiv 0 \pmod{N}$ and

$$N^{z_0} \cdot \prod_{i=1}^{n} p_i^{z_i\sigma^*(f,\mathfrak{p}_i)} > \|f\|_{2,s}^{\deg h} \cdot \|h\|_{2,s}^{\deg f} \tag{4.3}$$

will divide h (over $\mathbb{Q}$). The parameters $z_1, \ldots, z_n$ therefore allow the contributions of the roots of f modulo p_i to be weighted. Weighting the contributions is necessary since roots modulo large primes naturally contribute more to the product $\prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)}$, whereas primes modulo small primes contribute more to the root properties of f (cf. Section 2.1.2). Thus, a careful selection of the parameters $z_1, \ldots, z_n$ will, at least in theory, enable this bias to be corrected. In addition, since attention is limited in this setting to f with $\sigma^*(f, \langle N, x - m\rangle) > 0$, a careful selection of the corresponding parameter z_0 may be used to weaken the degree and size requirements on the polynomial h.

The inequality (4.3) shows that this approach to polynomial generation favours finding polynomials with small degree and skewed 2-norm. It follows that the approach naturally lends itself to nonlinear polynomial selection, where the degree and coefficient size of the polynomials considered is inherently smaller than for linear selection. In particular, the approach lends itself most readily to generating pairs of number field sieve polynomials with equal degree and, less favourably, pairs with degrees differing by one. In either of these cases, it is proposed that polynomial pairs are constructed from those factors f of h with $f(m) \equiv 0 \pmod N$. However, as with existing methods of polynomial generation, the proposed method requires a brute-force search to be performed over the parameter m. Therefore, in order to be competitive with existing methods, an extremely efficient realisation of Steps 1–3 above is required.

The approach developed in this section is applied in Section 4.4. However, before proceeding further, the validity of Lemma 4.1.1 must be established.

4.2 Divisibility Properties of Univariate Resultants

It is well-known that for two integer polynomials f_1 and f_2, a prime p divides $\mathrm{Res}(f_1, f_2)$ whenever f_1 and f_2 share a common root modulo p. More generally, p divides $\mathrm{Res}(f_1, f_2)$ whenever f_1 and f_2 share a common factor modulo p. These two facts were certainly known to Sylvester [162] and his contemporaries. Using p-adic methods, Konyagin and Shparlinski [96, Lemma 5.3] showed that p^μ divides the resultant whenever f_1 and f_2 share μ roots modulo p such that the roots of one polynomial are simple. They then asked [96, Question 5.4] if it were possible to remove the condition that the roots of one polynomial are simple. In response to this question, Gomez et al. [63] showed that if $\mathbb{A}$ is a unique factorisation domain, then a prime $p \in \mathbb{A}$ divides the resultant of two polynomials $f_1, f_2 \in \mathbb{A}[x] \setminus \langle p \rangle$ with multiplicity at least the degree of the gcd of their reductions modulo p. Their result answered the question raised by Konyagin and Shparlinski.

Although univariate resultants have been extensively studied, more is known about the divisibility properties of multivariate resultants than in the univariate case. In particular, Jouanolou [85, Section 6.2] proved the following:

Theorem 4.2.1. *Let $\mathbb{A}$ be an integral domain and $f_1, \ldots, f_n, g_1, \ldots, g_n \in \mathbb{A}[x_1, \ldots, x_n]$ be nonconstant homogeneous polynomials. Assume nonnegative integers $\mu_1, \ldots, \mu_n$ are given such that*

$$f_i \in \langle g_1, \ldots, g_n \rangle^{\mu_i} \subseteq \mathbb{A}[x_1, \ldots, x_n], \quad \text{for } 1 \leq i \leq n.$$

Then $\operatorname{Res}(g_1, \ldots, g_n)^{\mu_1 \cdots \mu_n}$ *divides* $\operatorname{Res}(f_1, \ldots, f_n)$ *in* $\mathbb{A}$.

Unfortunately, the obvious analogue of Theorem 4.2.1 for univariate resultants does not hold:

Example 4.2.2. The nontrivial ideal $\langle 2x^2 - 2, 2x - 1 \rangle \subset \mathbb{Z}[x]$ contains the linear polynomial $x - 2$. However, $\operatorname{Res}(x - 2, 2x - 1) = 3$ and $\operatorname{Res}(2x^2 - 2, 2x - 1) = 6$.

In the remainder of this section, the validity of Lemma 4.1.1 is established, thus improving upon the result of Gomez et al. and helping to close the gap between the known divisibility properties of univariate and multivariate resultants. In Section 4.2.1, the definition of the resultant of two univariate polynomials is extended to arbitrary integral domains. Then properties of resultants that are relevant to the remainder of the section (and the thesis) are discussed. In Section 4.2.2, an analogue of Theorem 4.2.1 is deduced for generic univariate polynomials and used to prove Lemma 4.1.1.

4.2.1 Definition and Properties of Resultants

The resultant of two polynomials f_1 and f_2 with coefficients in an integral domain $\mathbb{A}$, denoted $\operatorname{Res}(f_1, f_2)$, is an integer polynomial in the coefficients of f_1 and f_2 which vanishes if and only if the polynomials share a common root in a field containing $\mathbb{A}$. Resultants provide a tool for eliminating a variable between two polynomials and, as a result, play an important role in the field of computational algebraic geometry. Examples of resultants appear as far back as the late 1600's in the work of Leibniz [101]. There a method for computing the resultant of two quintic polynomials with the Euclidean algorithm was presented. The development of the theory of resultants can be attributed to the work of Euler [57], Bézout [20], Jacobi [80] and Cayley [34] (see [166, 168] for historical discussions). Here, Sylvester's [161] method for computing the resultant of two univariate polynomials is described.

Let $f_i = \sum_{j=0}^{m_i} u_{i,j} x^j$ be a non-constant polynomial with coefficients in an integral domain $\mathbb{A}$ and $u_{i,m_i} \neq 0$, for $i = 1, 2$. Then the *Sylvester matrix* of f_1 and f_2, denoted $\operatorname{Syl}(f_1, f_2)$, is defined to be the $(m_1 + m_2) \times (m_1 + m_2)$ matrix

$$\operatorname{Syl}(f_1, f_2) = \begin{pmatrix} u_{1,m_1} & u_{1,m_1-1} & \cdots & \cdots & u_{1,0} & & & \\ & u_{1,m_1} & u_{1,m_1-1} & \cdots & \cdots & u_{1,0} & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & u_{1,m_1} & u_{1,m_1-1} & \cdots & \cdots & u_{1,0} \\ u_{2,m_2} & u_{2,m_2-1} & \cdots & \cdots & u_{2,0} & & & \\ & u_{2,m_2} & u_{2,m_2-1} & \cdots & \cdots & u_{2,0} & & \\ & & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & u_{2,m_2} & u_{2,m_2-1} & \cdots & \cdots & u_{2,0} \end{pmatrix} \tag{4.4}$$

where there are m_2 rows containing the $u_{1,j}$, m_1 rows containing the $u_{2,j}$, and all empty entries are 0. Then the *resultant* of f_1 and f_2 is equal to the determinant of the Sylvester matrix $\mathrm{Syl}(f_1, f_2)$. For nonzero $u_1, u_2 \in \mathbb{A}$, the definition of the resultant is extended by setting

$$\mathrm{Res}(u_1, f_1) = \mathrm{Res}(f_1, u_1) = u_1^{m_1} \quad \text{and} \quad \mathrm{Res}(u_1, u_2) = 1.$$

Finally, for all $f \in \mathbb{A}[x]$, define $\mathrm{Res}(0, f) = \mathrm{Res}(f, 0) = 0$. When $\mathbb{A}$ is a polynomial ring, the notation $\mathrm{Res}_x(f_1, f_2)$ may be used to emphasise the polynomial variable.

Generalisations of resultants to multivariate polynomials were introduced in the works of Euler, Sylvester, Cayley and Macaulay and their contemporaries [166, p. 186]. Multivariate resultants are defined for $n \geq 2$ homogeneous polynomials in n variables. There are several formulations of multivariate resultants (see, for example, [89]). However, the resultant is uniquely determined by its properties (see [165, Chapter 11] or [61, Chapter 13]). For $n = 2$, the resultant of homogeneous polynomials $f_i = \sum_{j=0}^{m_i} u_{i,j} x^j y^{m_i - j}$, for $i = 1, 2$, is given by the determinant of the Sylvester matrix $\mathrm{Syl}(f_1, f_2)$ defined in (4.4). Resultants of polynomials in three or more variables are not considered here.

Throughout, the following well-known properties of univariate resultants are used:

Lemma 4.2.3 (Properties of Resultants)**.** Let $\mathbb{A}$ be an integral domain and $f_i = \sum_{j=0}^{m_i} u_{i,j} x^j \in \mathbb{A}[x]$ with $u_{i,m_i} \neq 0$ and $m_i \geq 1$, for $i = 1, 2$. Then the following properties holds:

(4.5) If the coefficients $u_{i,j}$ are algebraically independent indeterminates over $\mathbb{Z}$, then $\mathrm{Res}(f_1, f_2)$ is irreducible as an element of $\mathbb{Z}[u_{1,0}, \ldots, u_{1,m_1}, u_{2,0}, \ldots, u_{2,m_2}]$.

(4.6) $\mathrm{Res}(f_1, f_2) = 0$ if and only if f_1 and f_2 have a common root in a field containing $\mathbb{A}$.

(4.7) $\mathrm{Res}(f_1, f_2) = (-1)^{m_1 m_2} \cdot \mathrm{Res}(f_2, f_1)$.

(4.8) $\mathrm{Res}(f_1(x + y), f_2(x + y)) = \mathrm{Res}(f_1, f_2)$, for all $y \in \mathbb{A}$.

(4.9) If $f_3 \in \mathbb{A}[x]$, then $\mathrm{Res}(f_1, f_2 f_3) = \mathrm{Res}(f_1, f_2) \cdot \mathrm{Res}(f_1, f_3)$.

(4.10) $\mathrm{Res}(f_1, f_2)$ belongs to the ideal $\langle f_1, f_2 \rangle \cap \mathbb{A}$.

(4.11) Let $\mathbb{B}$ be an integral domain and $\varphi : \mathbb{A} \to \mathbb{B}$ a ring homomorphism. Let $\tilde{\varphi} : \mathbb{A}[x] \to \mathbb{B}[x]$ be the induced homomorphism given by $\tilde{\varphi}(\sum_i a_i x^i) = \sum_i \varphi(a_i) x^i$. If $\deg \tilde{\varphi}(f_1) = m_1$ and $\deg \tilde{\varphi}(f_2) = k$, $0 \leq k \leq m_2$, then

$$\varphi(\mathrm{Res}(f_1, f_2)) = \varphi(u_{1,m_1})^{m_2 - k} \cdot \mathrm{Res}(\tilde{\varphi}(f_1), \tilde{\varphi}(f_2)).$$

A proof of property (4.5) in Lemma 4.2.3 is provided by Macaulay [112, p. 5]. Proofs of the remaining properties stated in the lemma are provided by Apéry and Jouanolou [8].

4.2.2 Proof of Lemma 4.1.1

In this section, Lemma 4.1.1 is established. However, the majority of the section is devoted to proving the following generalisation of a result of Gomez et al. [63, Theorem 1], from which the lemma is obtained as a consequence:

Theorem 4.2.4. *Let $\mathbb{A}$ be an integral domain, $h \in \mathbb{A}$ be nonzero, and $g_1, \ldots, g_n \in \mathbb{A}[x]$ be nonconstant with unit leading coefficients. Suppose there exist polynomials $f_1, f_2 \in \mathbb{A}[x]$ with $f_1 \notin \langle h \rangle$ and $f_2 \notin \langle h \rangle$. Then given nonnegative integers $\mu_{1,1}, \ldots, \mu_{1,n}, \mu_{2,1}, \ldots, \mu_{2,n}$ such that*

$$f_i \in \langle g_1, h \rangle^{\mu_{i,1}} \cdots \langle g_n, h \rangle^{\mu_{i,n}} \subseteq \mathbb{A}[x], \quad \textit{for } i = 1, 2,$$

it follows that $\prod_{k=1}^{n} \operatorname{Res}(g_k, h)^{\mu_{1,k}\mu_{2,k}}$ divides $\operatorname{Res}(f_1, f_2)$ in $\mathbb{A}$.

The proof of Theorem 4.2.4 provided in this section employs generic polynomials: a univariate polynomial f is said to be *complete* whenever each monomial of degree at most $\deg f$ appears in f with a nonzero coefficient; it is said to be *generic* if it is complete and has coefficients that are algebraically independent indeterminates over $\mathbb{Z}$. The use of generic polynomials can aid in the derivation of universal properties of resultants. To help illustrate this observation, and to clarify what is meant by "universal", consider generic polynomials $\bar{f}_i = \sum_{j=0}^{m_i} \bar{u}_{i,j}\, x^j$, for $i = 1, 2$. Define $\mathbb{U} = \mathbb{Z}[\bar{u}_{i,j} \mid 0 \leq j \leq m_i, i = 1, 2]$, so that the polynomials $\bar{f}_i$ belong to the ring $\mathbb{U}[x]$. Let $\mathbb{A}$ be an integral domain and suppose polynomials $f_1, f_2 \in \mathbb{A}[x]$, of degree m_1 and m_2 respectively, are given:

$$f_i = \sum_{j=0}^{m_i} u_{i,j}\, x^j \in \mathbb{A}[x], \quad \text{for } i = 1, 2.$$

Then there exists a homomorphism $\varphi : \mathbb{U} \to \mathbb{A}$ defined by $\bar{u}_{i,j} \mapsto u_{i,j}$ corresponding to the *specialisation* of the polynomials $\bar{f}_i$ to the polynomials f_i. Therefore, (4.11) implies that $\operatorname{Res}(f_1, f_2) = \varphi\left(\operatorname{Res}(\bar{f}_1, \bar{f}_2)\right)$. As a result, a property of the resultant of two generic polynomials that is preserved under homomorphism is universal in the sense that it will also hold for any specialisation of the coefficients. In particular, this observation applies to the divisibility properties of resultants, thus motivating the use of generic polynomials in the proof of Theorem 4.2.4. The importance of ensuring each generic polynomial is specialised to a polynomial of equal degree must be emphasized: if $u_{1,m_1} = 0$ or $u_{2,m_2} = 0$ in the above example, then (4.11) implies that $\varphi\left(\operatorname{Res}(\bar{f}_1, \bar{f}_2)\right)$ and $\operatorname{Res}(f_1, f_2)$ are not necessarily equal. As a consequence, extreme care is taken whenever (4.11) is applied in this section.

A series of lemmas is now used to establish an appropriate generic polynomial analogue of Theorem 4.2.4. The last of these lemmas is then used to prove the theorem. Throughout this section, for generic polynomials $g_i = \sum_{j=0}^{d_i} v_{i,j} x^j$, $1 \leq i \leq k$, the ring $\mathbb{Z}[v_{i,j} \mid 1 \leq i \leq k, 0 \leq j \leq d_j]$ will often be denoted by $\mathbb{Z}[\{\text{coeff. of } g_1, \ldots, g_k\}]$.

Lemma 4.2.5. Let g_1 and g_2 be non-constant generic polynomials with distinct coefficients. Let μ_1, μ_2 be nonnegative integers and define polynomials f_1 and f_2 by

$$f_i = \sum_{j=0}^{\mu_i} a_{i,j}\, g_1^j\, g_2^{\mu_i - j}, \quad \text{for } i = 1, 2,$$

where $a_{i,j}$ is either a generic polynomial or 0, for $1 \le i \le 2$, $0 \le j \le \mu_i$; and the coefficients of g_1, g_2 and the nonzero $a_{i,j}$ are all distinct. Then $\operatorname{Res}(g_1, g_2)^{\mu_1\mu_2}$ divides $\operatorname{Res}(f_1, f_2)$ in $\mathbb{Z}[\{\text{coeff. of } a_{i,j}, g_i\}]$.

Proof. Assume that $\operatorname{Res}(f_1, f_2) \neq 0$, otherwise the lemma holds trivially. Let F_i denote the homogenisation of f_i, i.e., $F_i(x, y) = y^{\deg f_i} f_i(x/y)$, for $i = 1, 2$. Similarly, let G_1 and G_2 be the respective homogenisations of g_1 and g_2. Then $\operatorname{Res}(f_1, f_2) = \operatorname{Res}(F_1, F_2)$ and $\operatorname{Res}(g_1, g_2) = \operatorname{Res}(G_1, G_2)$ by definition. With $\mathbb{U} = \mathbb{Z}[\{\text{coeff. of } a_{i,j}, g_i\}]$, the lemma is therefore equivalent to the statement

$$\operatorname{Res}(G_1, G_2)^{\mu_1\mu_2} \text{ divides } \operatorname{Res}(F_1, F_2) \text{ in } \mathbb{U}. \tag{4.12}$$

Let $g_i = \sum_{j=0}^{n_i} v_{i,j}\, x^j$, for $i = 1, 2$. Then $a_{i,j}\, g_1^j\, g_2^{\mu_i - j}$ is homogeneous of degree j in the coefficient $v_{1,0}, \ldots, v_{1,n_1}$ and homogeneous of degree $\mu_i - j$ in the coefficients $v_{2,0}, \ldots, v_{2,n_2}$. As the coefficients of g_1, g_2 and the nonzero $a_{i,j}$ are all distinct, it follows that there cannot be any cancellation between the terms $a_{i,j}\, g_1^j\, g_2^{\mu_i - j}$ and $a_{i,k}\, g_1^k\, g_2^{\mu_i - k}$ of the polynomial f_i, for $0 \le j < k \le \mu_i$. Consequently,

$$\deg f_i \ge \deg a_{i,j} + j \cdot \deg g_1 + (\mu_i - j) \cdot \deg g_2,$$

for $1 \le i \le 2$, $0 \le j \le \mu_i$. Hence,

$$F_i = y^{\deg f_i} \cdot \sum_{j=0}^{\mu_i} a_{i,j}(x/y)\, g_1(x/y)^j\, g_2(x/y)^{\mu_i - j} = \sum_{j=0}^{\mu_i} b_{i,j}\, G_1^j\, G_2^{\mu_i - j}, \quad \text{for } i = 1, 2,$$

where the coefficient polynomials $b_{i,j} \in \mathbb{U}[x, y]$ are homogeneous. That is, $F_i \in \langle G_1, G_2 \rangle^{\mu_i}$ in $\mathbb{U}[x, y]$, for $i = 1, 2$. The polynomials F_1, F_2, G_1 and G_2 are non-constant. Hence, (4.12) is obtained by applying Theorem 4.2.1 (with $\mathbb{A} = \mathbb{U}$). □

Example 4.2.6. Let $g_1 = v_{1,1}\, x + v_{1,0}$ and $g_2 = v_{2,2}\, x^2 + v_{2,1}\, x + v_{2,0}$, where the $v_{i,j}$ are algebraically independent indeterminates over $\mathbb{Z}$. Then

$$\operatorname{Res}(g_1, g_2) = v_{1,1}^2\, v_{2,0} - v_{1,0}\, v_{1,1}\, v_{2,1} + v_{1,0}^2\, v_{2,2}.$$

Let $f_1 = a_{1,2}\, g_1^2 + a_{1,1}\, g_1\, g_2 + a_{1,0}\, g_2^2$ and $f_2 = a_{2,1}\, g_1 + a_{2,0}\, g_2$, for indeterminates $a_{i,j}$. Then

$$\operatorname{Res}(f_1, f_2) = v_{2,2}^2 \left(a_{1,2}\, a_{2,0}^2 - a_{1,1}\, a_{2,0}\, a_{2,1} + a_{1,0}\, a_{2,1}^2\right)^2 \left(v_{1,1}^2\, v_{2,0} - v_{1,0}\, v_{1,1}\, v_{2,1} + v_{1,0}^2\, v_{2,2}\right)^2.$$

Thus $\operatorname{Res}(g_1, g_2)^2$ divides $\operatorname{Res}(f_1, f_2)$ in $\mathbb{Z}[a_{i,j}, v_{i,j}]$, as expected.

The following example shows that Lemma 4.2.5 does not hold without modification if the polynomials g_1 and g_2 are permitted to be constant:

Example 4.2.7. Let $g_1 = v_{1,2}\,x^2 + v_{1,1}\,x + v_{1,0}$ and $g_2 = h$, where h and the $v_{i,j}$ are algebraically independent indeterminates over $\mathbb{Z}$. Then $\operatorname{Res}(g_1, g_2) = h^2$. Let $f_1 = a_{1,1}\,g_1\,g_2 + a_{1,0}\,g_2^2$ and $f_2 = a_{2,1}\,g_1\,g_2 + a_{2,0}\,g_2^2$, for indeterminates $a_{i,j}$. Then

$$\operatorname{Res}(f_1, f_2) = h^6\,v_{1,2}^2\,(a_{1,1}\,a_{2,0} - a_{1,0}\,a_{2,1})^2\,.$$

Thus, $\operatorname{Res}(g_1, g_2)^4$ does not divide $\operatorname{Res}(f_1, f_2)$ in $\mathbb{Z}[a_{i,j}, v_{1,j}][h]$.

Lemma 4.2.8. Let h be an indeterminate over $\mathbb{Z}$ and g be a non-constant generic polynomial with coefficients distinct from h. Let μ_1, μ_2 be nonnegative integers and define polynomials f_1 and f_2 by

$$f_i = \sum_{j=0}^{\mu_i} a_{i,j} g^j h^{\mu_i - j}, \quad \text{for } i = 1, 2,$$

where $a_{i,j}$ is either a generic polynomial or 0, for $1 \le i \le 2$, $0 \le j \le \mu_i$; the coefficients of g, h and the nonzero $a_{i,j}$ are all distinct; and $a_{i,\mu_i} \ne 0$, for $i = 1, 2$. Then $\operatorname{Res}(g, h)^{\mu_1\mu_2}$ divides $\operatorname{Res}(f_1, f_2)$ in $\mathbb{Z}[\{\text{coeff. of } a_{i,j}, g, h\}]$.

Proof. Assume that $\mu_1\mu_2 \ne 0$, otherwise the lemma holds trivially. Define $\mathbb{U} = \mathbb{Z}[\{\text{coeff. of } a_{i,j}, g, h\}]$ and let $g = \sum_{j=0}^{n} v_j x^j$, where $v_n \ne 0$. Since $g \cdot g^j h^{\mu_i - j} = h \cdot g^{j+1} h^{\mu_i - j - 1}$, the polynomial f_i may be written in the form

$$f_i = \sum_{j=0}^{\mu_i} b_{i,j} g^j h^{\mu_i - j},$$

where $b_{i,j} \in \mathbb{U}[v_n^{-1}][x]$, for $0 \le j \le \mu_i$; and $\deg b_{i,j} \le (n-1)(\mu_i - j)$, for $0 \le j < \mu_i$. The condition $a_{i,\mu_i} \ne 0$ implies that $\deg f_i \ge n\mu_i$, for $i = 1, 2$. Therefore, by comparing degrees, it follows that $b_{i,\mu_i} \ne 0$, for $i = 1, 2$.

For each polynomial $b_{i,j}$, let $\bar{b}_{i,j}$ be a generic polynomial of equal degree, if $b_{i,j} \ne 0$; and $\bar{b}_{i,j} = 0$, otherwise. Furthermore, impose the requirement that the coefficients of g, h and the nonzero $\bar{b}_{i,j}$ are algebraically independent over $\mathbb{Z}$. Define $\overline{\mathbb{U}} = \mathbb{Z}[\{\text{coeff. of } \bar{b}_{1,0}, \ldots, \bar{b}_{1,\mu_1}, \bar{b}_{2,0}, \ldots, \bar{b}_{2,\mu_2}, g, h\}]$ and let $y \notin \overline{\mathbb{U}}$ be an indeterminate. Finally, define polynomials

$$\bar{f}_i = \sum_{j=0}^{\mu_i} \bar{b}_{i,j} g^j (h + yx)^{\mu_i - j} \in \overline{\mathbb{U}}[y][x], \quad \text{for } i = 1, 2. \tag{4.13}$$

Then there exists an evaluation homomorphism $\varphi : \overline{\mathbb{U}}[y] \to \mathbb{U}[v_n^{-1}]$, with induced homomorphism $\tilde{\varphi} : \overline{\mathbb{U}}[y][x] \to \mathbb{U}[v_n^{-1}][x]$, such that $\varphi(y) = 0$; $\tilde{\varphi}(g) = g$, $\tilde{\varphi}(h + yx) = h$; and $\tilde{\varphi}(\bar{f}_1) = f_1$, $\tilde{\varphi}(\bar{f}_2) = f_2$. Moreover, the observation that $b_{i,\mu_i} \ne 0$ combined with the inequalities $\deg b_{i,j} \le (n-1)(\mu_i - j)$, for

$0 \leq j < \mu_i$, imply that $\deg \tilde{\varphi}(\bar{f}_1) = \deg \bar{f}_1$ and $\deg \tilde{\varphi}(\bar{f}_2) = \deg \bar{f}_2$. Consequently, (4.11) implies that

$$\varphi(\mathrm{Res}(\bar{f}_1, \bar{f}_2)) = \mathrm{Res}(\tilde{\varphi}(\bar{f}_1), \tilde{\varphi}(\bar{f}_2)) = \mathrm{Res}(f_1, f_2).$$

From (4.13) and Lemma 4.2.5, it follows that

$$\mathrm{Res}(h + yx, g)^{\mu_1 \mu_2} \text{ divides } \mathrm{Res}(\bar{f}_1, \bar{f}_2) \text{ in } \overline{\mathbb{U}}[y].$$

Thus

$$\varphi\left(\mathrm{Res}(h + yx, g)\right)^{\mu_1 \mu_2} \text{ divides } \mathrm{Res}(f_1, f_2) \text{ in } \mathbb{U}[v_n^{-1}].$$

Properties (4.7) and (4.11) imply that $\varphi\left(\mathrm{Res}(h + yx, g)\right) = (-1)^n v_n \mathrm{Res}(g, h)$. Therefore, $\mathrm{Res}(g, h)^{\mu_1 \mu_2}$ divides $\mathrm{Res}(f_1, f_2)$ in $\mathbb{U}[v_n^{-1}]$. However, $\mathrm{Res}(f_1, f_2) \in \mathbb{U}$ and $h \nmid v_n$. Hence, $\mathrm{Res}(g, h)^{\mu_1 \mu_2}$ must divide $\mathrm{Res}(f_1, f_2)$ in $\mathbb{U}$. □

Lemma 4.2.9. Let $h_1, \ldots, h_n$ be algebraically independent indeterminates over $\mathbb{Z}$; $g_1 \ldots g_n$ be non-constant generic polynomials with algebraically independent coefficients over $\mathbb{Z}[h_1, \ldots, h_n]$. For non-negative integers $\mu_{1,1}, \ldots, \mu_{1,n}, \mu_{2,1}, \ldots, \mu_{2,n}$, define polynomials f_1 and f_2 by

$$f_i = \sum_{\substack{0 \leq j_1 \leq \mu_{i,1} \\ \cdots \\ 0 \leq j_n \leq \mu_{i,n}}} a_{i,j_1,\ldots,j_n} \cdot g_1^{j_1} \cdots g_n^{j_n} \cdot h_1^{\mu_{i,1} - j_1} \cdots h_n^{\mu_{i,n} - j_n}, \quad \text{for } i = 1, 2.$$

where $a_{i,j_1,\ldots,j_n}$ is either a generic polynomial or 0; $a_{i,\mu_{i,1},\ldots,\mu_{i,n}} \neq 0$, for $i = 1, 2$; and the coefficients of $h_1, \ldots, h_n, g_1, \ldots, g_n$ and the nonzero $a_{i,j_1,\ldots,j_n}$ are all distinct. Then $\prod_{k=1}^{n} \mathrm{Res}(g_k, h_k)^{\mu_{1,k} \mu_{2,k}}$ divides $\mathrm{Res}(f_1, f_2)$, in $\mathbb{Z}[\{\text{coeff. of } a_{i,j_1,\ldots,j_n}, g_1 \ldots g_n, h_1, \ldots, h_n\}]$.

Proof. For simplicity, let $\mathbb{U} = \mathbb{Z}[\{\text{coeff. of } a_{i,j_1,\ldots,j_n}, g_1 \ldots g_n, h_1, \ldots, h_n\}]$. By assumption, $h_1, \ldots, h_n$ are algebraically independent indeterminates over $\mathbb{Z}$ and $\mathrm{Res}(g_k, h_k) = h_k^{\deg g_k}$, for $1 \leq k \leq n$. Therefore, it is sufficient to show that

$$\mathrm{Res}(g_k, h_k)^{\mu_{1,k} \mu_{2,k}} \text{ divides } \mathrm{Res}(f_1, f_2) \text{ in } \mathbb{U}, \text{ for } 1 \leq k \leq n. \tag{4.14}$$

Here, (4.14) is shown to hold for $k = 1$ only. The remaining cases follow in a similar fashion.

Assume that $\mu_{1,1} \mu_{2,1} \neq 0$, otherwise (4.14) holds trivially for $k = 1$. Consequently, f_1 and f_2 are non-constant since $\deg g_1 \geq 1$ and $a_{i,\mu_{i,1},\ldots,\mu_{i,n}} \neq 0$, for $i = 1, 2$. Define

$$b_{i,j} = \sum_{\substack{0 \leq j_2 \leq \mu_{i,2} \\ \cdots \\ 0 \leq j_n \leq \mu_{i,n}}} a_{i,j,j_2,\ldots,j_n} \cdot g_2^{j_2} \cdots g_n^{j_n} \cdot h_2^{\mu_{i,2} - j_2} \cdots h_n^{\mu_{i,n} - j_n},$$

for $1 \le i \le 2$, $0 \le j \le \mu_{i,1}$. Then

$$f_i = \sum_{j=0}^{\mu_{i,1}} b_{i,j}\, g_1^j\, h_1^{\mu_{i,1}-j}, \quad \text{for } i = 1, 2.$$

Moreover, $b_{1,\mu_{1,1}}$ and $b_{2,\mu_{2,1}}$ are both nonzero since $a_{i,\mu_{i,1},\ldots,\mu_{i,n}} \neq 0$, for $i = 1, 2$. For each polynomial $b_{i,j}$, let $\bar{b}_{i,j}$ be a generic polynomial of equal degree, if $b_{i,j} \neq 0$; and $\bar{b}_{i,j} = 0$, otherwise. Furthermore, impose the requirement that the coefficients of the g_1, h_1 and the nonzero $\bar{b}_{i,j}$ are algebraically independent over $\mathbb{Z}$. Define $\overline{\mathbb{U}} = \mathbb{Z}[\{\text{coeff. of } \bar{b}_{1,0}, \ldots, \bar{b}_{1,\mu_{1,1}}, \bar{b}_{2,0}, \ldots, \bar{b}_{2,\mu_{2,1}}, g_1, h_1\}]$ and polynomials

$$\bar{f}_i = \sum_{j=0}^{\mu_{i,1}} \bar{b}_{i,j}\, g_1^j\, h_1^{\mu_{i,1}-j} \in \overline{\mathbb{U}}[x], \quad \text{for } i = 1, 2. \tag{4.15}$$

Then there exists a homomorphism $\varphi : \overline{\mathbb{U}} \to \mathbb{U}$, with induced homomorphism $\tilde{\varphi} : \overline{\mathbb{U}}[x] \to \mathbb{U}[x]$, such that $\tilde{\varphi}(g_1) = g_1$, $\tilde{\varphi}(h_1) = h_1$; and $\tilde{\varphi}(\bar{f}_1) = f_1$, $\tilde{\varphi}(\bar{f}_2) = f_2$. As the coefficients of $g_1 \ldots g_n, h_1, \ldots, h_n$ and the nonzero $a_{i,j_1,\ldots,j_n}$ are algebraically independent, it follows that $\deg \bar{f}_i = \deg f_i$, for $i = 1, 2$. Therefore, (4.11) implies that

$$\varphi(\operatorname{Res}(\bar{f}_1, \bar{f}_2)) = \operatorname{Res}(\tilde{\varphi}(\bar{f}_1), \tilde{\varphi}(\bar{f}_2)) = \operatorname{Res}(f_1, f_2).$$

In addition, $\varphi(\operatorname{Res}(g_1, h_1)) = \operatorname{Res}(g_1, h_1)$, since $\varphi(h_1) = h_1$. Hence, (4.14) will hold for $k = 1$ if

$$\operatorname{Res}(g_1, h_1)^{\mu_{1,1}\mu_{2,1}} \text{ divides } \operatorname{Res}(\bar{f}_1, \bar{f}_2) \text{ in } \overline{\mathbb{U}}. \tag{4.16}$$

Now $\bar{b}_{1,\mu_1}$ and $\bar{b}_{2,\mu_2}$ are both nonzero since $b_{i,\mu_{i,1}} \neq 0$, for $i = 1, 2$. Therefore, (4.15) and Lemma 4.2.8 imply (4.16). □

Lemma 4.2.9 is sufficient to establish Theorem 4.2.4.

Proof of Theorem 4.2.4. Assume that $\operatorname{Res}(f_1, f_2) \neq 0$, otherwise the lemma holds trivially. Without loss of generality, it is assumed that $\mu_{1,k}\mu_{2,k} \neq 0$, for $1 \le k \le n$. Define sets

$$J_i = \{(j_1, \ldots, j_n) \in \mathbb{Z}^n \mid 0 \le j_1 \le \mu_{i,1}, \ldots, 0 \le j_n \le \mu_{i,n}\}, \quad \text{for } i = 1, 2.$$

Define a well-ordering $\prec_i$ on J_i by $(j_1, \ldots, j_n) \prec_i (j_1', \ldots, j_n')$ if and only if the left most nonzero entry of the vector $(j_1' - j_1, \ldots, j_n' - j_n)$ is positive. Then $(\mu_{i,1}, \ldots, \mu_{i,n})$ is the greatest element of J_i under $\prec_i$, for $i = 1, 2$.

It follows from the definition of the product of two ideals, that there exist polynomials $a_{i,j_1,\ldots,j_n} \in \mathbb{A}[x]$ such that

$$f_i = \sum_{(j_1,\ldots,j_n) \in J_i} a_{i,j_1,\ldots,j_n} \cdot g_1^{j_1} \cdots g_n^{j_n} \cdot h^{(\mu_{i,1}-j_1)+\ldots+(\mu_{i,n}-j_n)}, \quad \text{for } i = 1, 2. \tag{4.17}$$

Given a vector $(j_1, \ldots, j_n) \in J_i \setminus \{(\mu_{i,1}, \ldots, \mu_{i,n})\}$, the coefficient polynomial $a_{i,j_1,\ldots,j_n}$ is henceforth referred to as *reduced* if there exists an index k, $1 \le k \le n$, such that $j_k \ne \mu_{i,k}$ and $\deg a_{i,j_1,\ldots,j_n} < \deg g_k$. It follows that

$$\deg a_{i,j_1,\ldots,j_n} \cdot g_1^{j_1} \cdots g_n^{j_n} < \deg g_1^{\mu_{i,1}} \cdots g_n^{\mu_{i,n}}, \tag{4.18}$$

for any $(j_1, \ldots, j_n) \in J_i \setminus \{(\mu_{i,1}, \ldots, \mu_{i,n})\}$ such that $a_{i,j_1,\ldots,j_n}$ is reduced.

Suppose there exists a vector $(j_1, \ldots, j_n) \in J_i \setminus \{(\mu_{i,1}, \ldots, \mu_{i,n})\}$ such that $a_{i,j_1,\ldots,j_n}$ is not reduced. Let k, $1 \le k \le n$, be some index such that $j_k \ne \mu_{i,k}$. Since $\mathbb{A}$ is an integral domain and the leading coefficient of g_k is a unit, there exist polynomials $q_{i,j_1,\ldots,j_n}, r_{i,j_1,\ldots,j_n} \in \mathbb{A}[x]$ such that $a_{i,j_1,\ldots,j_n} = q_{i,j_1,\ldots,j_n} g_k + r_{i,j_1,\ldots,j_n}$ and $\deg r_{i,j_1,\ldots,j_n} < \deg g_k$. Therefore, $a_{i,j_1,\ldots,j_n}$ is made to be reduced by subtracting $q_{i,j_1,\ldots,j_n} g_k$ from $a_{i,j_1,\ldots,j_n}$ and adding $q_{i,j_1,\ldots,j_n} h$ to $a_{i,j_1,\ldots,j_{k-1},j_k+1,j_{k+1},\ldots,j_n}$. After this procedure, the polynomial f_i remains unchanged. In addition, the choice of k guarantees that $(j_1, \ldots, j_{k-1}, j_k + 1, j_{k+1}, \ldots, j_n) \in J_i$, thus

$$(j_1, \ldots, j_n) \prec_i (j_1, \ldots, j_{k-1}, j_k + 1, j_{k+1}, \ldots, j_n).$$

Hence, the procedure just described can be applied repeatedly to the least element of $(j_1, \ldots, j_n) \in J_i \setminus \{(\mu_{i,1}, \ldots, \mu_{i,n})\}$ under $\prec_i$ such that $a_{i,j_1,\ldots,j_n}$ is not reduced until no such element remains. That is, it may be assumed without loss of generality that the coefficients $a_{i,j_1,\ldots,j_n}$ in (4.17) are reduced for all $(j_1, \ldots, j_n) \in J_i \setminus \{(\mu_{i,1}, \ldots, \mu_{i,n})\}$ and $i = 1, 2$.

For $i = 1, 2$, the coefficient $a_{i,\mu_{i,1},\ldots,\mu_{i,n}}$ in (4.17) must be nonzero, otherwise $f_i \in \langle h \rangle$. It follows that

$$\deg f_i \ge \deg a_{i,j_1,\ldots,j_n} \cdot g_1^{j_1} \cdots g_n^{j_n}, \quad \text{for all } (j_1, \ldots, j_n) \in J_i \text{ and } i = 1, 2. \tag{4.19}$$

In particular, (4.18) implies that equality holds if and only if $(j_1, \ldots, j_n) = (\mu_{i,1}, \ldots, \mu_{i,n})$. Consequently, f_1 and f_2 are non-constant. Let $\bar{h}_1, \ldots, \bar{h}_n$ be algebraically independent indeterminates over $\mathbb{Z}$. To each of the polynomials $g_1, \ldots, g_n$ and the nonzero $a_{i,j_1\ldots,j_n}$, assign respective generic polynomials $\bar{g}_1, \ldots, \bar{g}_n$ and $\bar{a}_{i,j_1\ldots,j_n}$ of equal degree and with coefficients that are algebraically independent over $\mathbb{Z}[\bar{h}_1, \ldots, \bar{h}_n]$. Additionally, for those $a_{i,j_1\ldots,j_n} = 0$, define $\bar{a}_{i,j_1\ldots,j_n} = 0$. Finally, define $\mathbb{U} = \mathbb{Z}[\{\text{coeff. of } \bar{h}_1, \ldots, \bar{h}_n, \bar{g}_1 \ldots \bar{g}_n, \bar{a}_{i,j_1,\ldots,j_n}\}]$ and polynomials

$$\bar{f}_i = \sum_{(j_1,\ldots,j_n)\in J_i} \bar{a}_{i,j_1,\ldots,j_n} \cdot \bar{g}_1^{j_1} \cdots \bar{g}_n^{j_n} \cdot \bar{h}_1^{\mu_{i,1}-j_1} \cdots \bar{h}_n^{\mu_{i,n}-j_n} \in \mathbb{U}[x], \quad \text{for } i = 1, 2.$$

Then (4.19) implies that $\deg \bar{f}_i = \deg f_i$, for $i = 1, 2$. Moreover, there exists a homomorphism $\varphi : \mathbb{U} \to \mathbb{A}$, with induced homomorphism $\tilde{\varphi} : \mathbb{U}[x] \to \mathbb{A}[x]$, such that $\tilde{\varphi}(\bar{h}_k) = h$, for $1 \le k \le n$; $\tilde{\varphi}(\bar{g}_k) = g_k$, for $1 \le k \le n$; and $\tilde{\varphi}(\bar{f}_i) = f_i$, for $i = 1, 2$. Hence, (4.11) implies that

$$\varphi(\operatorname{Res}(\bar{g}_k, \bar{h}_k)) = \operatorname{Res}(g_k, h), \text{ for } 1 \le k \le n; \quad \text{and} \quad \varphi(\operatorname{Res}(\bar{f}_1, \bar{f}_2)) = \operatorname{Res}(f_1, f_2). \tag{4.20}$$

As $a_{1,\mu_{1,1},\ldots,\mu_{1,n}}$ and $a_{2,\mu_{2,1},\ldots,\mu_{2,n}}$ are nonzero, it follows that $\bar{a}_{i,\mu_{i,1},\ldots,\mu_{i,n}} \neq 0$, for $i = 1, 2$. Therefore, Lemma 4.2.9 and the definition of $\bar{f}_1$ and $\bar{f}_2$ imply that $\prod_{k=1}^{n} \operatorname{Res}(\bar{g}_k, \bar{h}_k)^{\mu_{1,k}\mu_{2,k}}$ divides $\operatorname{Res}(\bar{f}_1, \bar{f}_2)$ in $\mathbb{U}$. Hence, (4.20) implies that $\prod_{k=1}^{n} \operatorname{Res}(g_k, h)^{\mu_{1,k}\mu_{2,k}}$ divides $\operatorname{Res}(f_1, f_2)$ in $\mathbb{A}$. □

Before ending the section with a proof of Lemma 4.1.1, Theorem 4.2.4 is now compared with the result of Gomez et al. [63, Theorem 1]:

Example 4.2.10. For any $p, k \in \mathbb{Z}$ with $k > 0$, let $f_1 = x(x-1)$ and $f_2 = (x - p^k)(x-2)$. Then $\operatorname{Res}(f_1, f_2) = 2p^k(p^k - 1)$. Gomez et al. [63, Example 2] showed that p divides $\operatorname{Res}(f_1, f_2)$, whenever p is an odd prime. The polynomials f_1 and f_2 are primitive and $f_1, f_2 \in \langle 2p^k, x\rangle$, for all $p \in \mathbb{Z}$. Therefore, for *any* $p \in \mathbb{Z}$, Theorem 4.2.4 implies that $\operatorname{Res}(2p^k, x) = 2p^k$ divides $\operatorname{Res}(f_1, f_2)$. Moreover,

$$f_2 = (x - p^k)(x-2) = (x - 1 + 1 - p^k)(x-2) = (x-1)(x-2) - (p^k - 1)(x-2).$$

It follows that $f_1, f_2 \in \langle p^k - 1, x - 1\rangle$, for all $p \in \mathbb{Z}$. Hence, Theorem 4.2.4 implies that $\operatorname{Res}(p^k - 1, x - 1) = p^k - 1$ divides $\operatorname{Res}(f_1, f_2)$, for all $p \in \mathbb{Z}$.

Proof of Lemma 4.1.1. Let f_1 and f_2 be nonzero integer polynomials such that f_1 is non-constant and primitive. The proof of the lemma is based on the following claim: if $h, r_1, \ldots, r_w$ are integers such that h is nonzero and $\gcd(r_j - r_k, h) = 1$, for $1 \le j < k \le w$, then

$$\prod_{k=1}^{w} h^{\sigma^*(f_1,\langle h, x - r_k\rangle)\sigma^*(f_2,\langle h, x-r_k\rangle)} \text{ divides } \operatorname{Res}(f_1, f_2) \text{ in } \mathbb{Z}. \tag{4.21}$$

Given the claim, Lemma 4.1.1 then follows from the definition of the ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ and the assumption that $p_i \nmid N$, for $1 \le i \le n$.

To simplify notation throughout the proof of the claim, define $\mu_{i,j} = \sigma^*(f_i, \langle h, x - r_j\rangle)$, for $1 \le i \le 2$, $1 \le j \le w$. If $f_2 \notin \langle h\rangle$, then the assumption that f_1 is primitive and Theorem 4.2.4 imply (4.21). Therefore, assume that $f_2 \in \langle h\rangle$. Then there exists a value of $t \ge 1$ and a polynomial $f_2^* \in \mathbb{Z}[x] \setminus \langle h\rangle$ such that $f_2 = h^t \cdot f_2^*$. As a result, (4.9) implies that

$$\operatorname{Res}(f_1, f_2) = \operatorname{Res}(f_1, h^t \cdot f_2^*) = \operatorname{Res}(f_1, h^t) \cdot \operatorname{Res}(f_1, f_2^*) = h^{t \deg f_1} \cdot \operatorname{Res}(f_1, f_2^*). \tag{4.22}$$

Fix an index k, $1 \le k \le w$, then $f_2 \in \langle h, x - r_k\rangle^{\mu_{2,k}}$. It follows that there exist polynomials $a_0, \ldots, a_{\mu_{2,k}} \in \mathbb{Z}[x]$ such that $f_2 = \sum_{j=0}^{\mu_{2,k}} a_j \cdot (x - r_k)^j h^{\mu_{2,k} - j}$. Furthermore, it may be assumed that the coefficients $a_0, \ldots, a_{\mu_{2,k}-1}$ are integers, since

$$(x - r_k) \cdot (x - r_k)^j h^{\mu_{2,k} - j} = h \cdot (x - r_k)^{j+1} h^{\mu_{2,k} - j - 1}.$$

From the observation that

$$f_2(x+r_k) = h^t \cdot f_2^*(x+r_k) = a_{\mu_{2,k}}(x+r_k) \cdot x^{\mu_{2,k}} + \sum_{j=0}^{\mu_{2,k}-1} \left(a_j h^{\mu_{2,k}-j}\right) \cdot x^j,$$

it is readily deduced that $h^{t-(\mu_{2,k}-j)}$ divides a_j, for $\mu_{2,k} - t \leq j \leq \mu_{2,k}$. Therefore, there exists a polynomial $a'_{\mu_{2,k}-t} \in \mathbb{Z}[x]$ such that

$$f_2 = h^t \cdot \left(a'_{\mu_{2,k}-t} \cdot (x-r_k)^{\mu_{2,k}-t} + \sum_{j=0}^{(\mu_{2,k}-t)-1} a_j \cdot (x-r_k)^j h^{(\mu_{2,k}-t)-j} \right)$$

It follows immediately that $f_2^* \in \langle h, x-r_k \rangle^{\mu_{2,k}-t}$. Moreover, since k was arbitrary and the ideals $\langle h, x-r_k \rangle$, for $1 \leq k \leq w$, are pairwise comaximal, it follows that $f_2^* \in \prod_{k=1}^{w} \langle h, x-r_k \rangle^{\mu_{2,k}-t}$. Therefore, Theorem 4.2.4 implies that

$$\prod_{k=1}^{w} h^{\mu_{1,k}(\mu_{2,k}-t)} \text{ divides } \operatorname{Res}(f_1, f_2^*) \text{ in } \mathbb{Z}. \tag{4.23}$$

Finally, from the assumption that $f_1 \notin \langle h \rangle$ and the inequality (4.19) in the proof of Theorem 4.2.4, it follows that

$$\deg f_1 \geq \mu_{1,1} + \ldots + \mu_{1,w}. \tag{4.24}$$

Hence, (4.22), (4.23) and (4.24) imply (4.21). □

4.3 Combinatorial Bounds on Polynomial Selection

A polynomial generation algorithm that is based on the approach of Section 4.1 is guaranteed to return *all* polynomials with sufficiently small coefficients and good non-projective root properties. Therefore, a necessary condition for such an algorithm to run in polynomial time is that only polynomially many polynomials are found. In this section, a purely combinatorial result is used to derive bounds on the existence of number field sieve polynomials with small coefficients and good non-projective root properties. Then a condition under which polynomial generation is combinatorially feasible is derived. To begin, the main combinatorial result used in this section, due to Guruswami [68, Theorem 7.10], is stated:

Lemma 4.3.1. Let $\Sigma_1, \ldots, \Sigma_n$ be finite nonempty sets and $\mathcal{C} \subseteq \Sigma_1 \times \ldots \times \Sigma_n$ be nonempty. Let vectors $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ contain positive real entries. Define $d(\mathcal{C})_{\boldsymbol{\alpha}}$ to be the minimum of $\sum_{i:x_i \neq y_i} \alpha_i$ over all pairs of distinct vectors $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \mathcal{C}$. Then, for any vector $(t_1, \ldots, t_n) \in \Sigma_1 \times \ldots \times \Sigma_n$ and $l > 0$, there exist at most l vectors $(x_1, \ldots, x_n) \in \mathcal{C}$ such that

$$\sum_{i:x_i=t_i} \beta_i \geq \sqrt{\left(\sum_{i=1}^{n} \alpha_i - \left(1 - \frac{1}{l}\right) d(\mathcal{C})_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\alpha_i}}.$$

By combining Lemma 4.3.1 with Lemma 4.1.2, the following combinatorial bound on the existence of number field sieve polynomials with small coefficients and good non-projective root properties is obtained:

Theorem 4.3.2. *Let C, s be positive reals with $C \geq 1$; d, m, N be integers with $d, N \geq 1$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. Given positive real weights $\beta_1, \ldots, \beta_n$ and a real number $l \geq 1$, there are at most $2l$ non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that*

$$\sum_{i=1}^{n} \sigma(f, \mathfrak{p}_i)\beta_i \geq \sqrt{\left(\left(1 - \frac{1}{l}\right) 2d \log C + \frac{1}{l} \sum_{i=1}^{n} \log p_i\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\log p_i}}. \tag{4.25}$$

Proof. For all $f \in \mathbb{Z}[x]$, $\sigma(f, \mathfrak{p}_i) = \sigma(-f, \mathfrak{p}_i)$, for $1 \leq i \leq n$. Therefore, it is sufficient to show that there are at most l polynomials that satisfy the conditions of the theorem and have positive leading coefficient. Let $\mathcal{M}$ be the set of all non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with positive leading coefficient that satisfy $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$. Then $\mathcal{M}$ is finite due to the degree and size requirements on its elements. For $1 \leq i \leq n$, define $\mathcal{E}_i : \mathcal{M} \to \mathcal{M} \cup \{0\}$ by $\mathcal{E}_i(f) = 0$, if $f \in \mathfrak{p}_i$; and $\mathcal{E}_i(f) = f$, otherwise. Finally, define $\mathcal{C} \subset (\mathcal{M} \cup \{0\})^n$ by

$$\mathcal{C} = \{(\mathcal{E}_1(f), \cdots, \mathcal{E}_n(f)) \mid f \in \mathcal{M}\}.$$

Then, applying Lemma 4.3.1 with $\boldsymbol{\alpha} = (\log p_1, \ldots, \log p_n)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ and $(t_1, \ldots, t_n) = (0, \ldots, 0)$, it follows that there are at most l vectors $(\mathcal{E}_1(f), \cdots, \mathcal{E}_n(f)) \in \mathcal{C}$ such that

$$\sum_{i:\mathcal{E}_i(f)=0} \beta_i \geq \sqrt{\left(\sum_{i=1}^{n} \log p_i - \left(1 - \frac{1}{l}\right) d(\mathcal{C})_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\log p_i}}, \tag{4.26}$$

where $d(\mathcal{C})_{\boldsymbol{\alpha}}$ is the minimum value of the sum $\sum_{i:\mathcal{E}_i(f_1)\neq\mathcal{E}_i(f_2)} \log p_i$, over all distinct pairs of vectors $(\mathcal{E}_1(f_1), \cdots, \mathcal{E}_n(f_1)), (\mathcal{E}_1(f_2), \cdots, \mathcal{E}_n(f_2)) \in \mathcal{C}$.

If $f_1, f_2 \in \mathcal{M}$ are distinct, then (4.6) implies that $\mathrm{Res}(f_1, f_2) \neq 0$. Consequently, Lemma 4.1.2 implies that

$$N \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f_1,\mathfrak{p}_i)\sigma^*(f_2,\mathfrak{p}_i)} \leq \|f_1\|_{2,s}^{\deg f_2} \cdot \|f_2\|_{2,s}^{\deg f_1} \leq C^{2d} \cdot N.$$

Therefore, if $(\mathcal{E}_1(f_1), \cdots, \mathcal{E}_n(f_1)), (\mathcal{E}_1(f_2), \cdots, \mathcal{E}_n(f_2)) \in \mathcal{C}$ are distinct vectors, then

$$\sum_{i:\mathcal{E}_i(f_1)\neq\mathcal{E}_i(f_2)} \log p_i = \sum_{i=1}^{n} (1 - \sigma(f_1, \mathfrak{p}_i)\sigma(f_2, \mathfrak{p}_i)) \log p_i \geq \sum_{i=1}^{n} \log p_i - 2d \log C.$$

Hence, $d(\mathcal{C})_{\boldsymbol{\alpha}} \geq \sum_{i=1}^{n} \log p_i - 2d \log C$.

For all $(\mathcal{E}_1(f), \cdots, \mathcal{E}_n(f)) \in \mathcal{C}$, the right hand side of (4.26) is $\sum_{i:\mathcal{E}_i(f)=0} \beta_i = \sum_{i=1}^{n} \sigma(f, \mathfrak{p}_i)\beta_i$. Therefore, if $f \in \mathcal{M}$ satisfies (4.25), then the corresponding vector $(\mathcal{E}_1(f), \cdots, \mathcal{E}_n(f)) \in \mathcal{C}$ satisfies (4.26). Hence, there exist at most l polynomials $f \in \mathcal{M}$ such that (4.25) holds. □

Remark 4.3.3. Guruswami remarked [68, p. 163] that a stronger bound than that provided by Lemma 4.3.1 can be obtained by taking into account the size of the alphabet Σ. Furthermore, Guruswami notes that for large alphabets, the difference between the bounds "becomes negligible". Therefore, it may be possible to tighten the bound in Theorem 4.3.2 by strengthening Lemma 4.3.1. However, the difference between the resulting bounds depends on the alphabet size, which, in the proof of Theorem 4.3.2, is equal to $|\mathcal{M}| + 1$, i.e., one more than the number of non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$. If this number is sufficiently small for small d and C, then it may be worthwhile further investigating this avenue.

Recall that the root properties of a number field sieve polynomial f can be quantified by the parameter $\alpha(f, y)$ (see Section 2.1.2). The contribution of non-projective roots to $\alpha(f, y)$ is approximated by the quantity $\bar{\alpha}(f, y)$, defined for all $f \in \mathbb{Z}[x]$ and $y > 0$ by

$$\bar{\alpha}(f, y) = \sum_{p \leq y} \left(1 - \frac{p}{p+1} \sum_{r=0}^{p-1} \sigma(f, \mathfrak{p}_{p,r})\right) \frac{\log p}{p-1}, \tag{4.27}$$

where the sum is over all primes $p \leq y$. The error of this approximation is determined by those primes $p \leq y$ that divide $\operatorname{disc}(f)$. Throughout this chapter, the quantity $\bar{\alpha}(f, y)$ is studied rather than $\alpha(f, y)$. The motivation for this departure is twofold. First, Lemma 4.1.2 is not sufficiently strong as to allow for projective root properties to be handled naturally throughout the chapter. Second, unlike the contribution of non-projective roots to $\alpha(f, y)$, the approximation $\bar{\alpha}(f, y)$ is given by a simple closed-form expression. Therefore, in the context of this chapter, it is reasonable to study $\bar{\alpha}(f, y)$.

The following corollary to Theorem 4.3.2 provides a bound on the number of non-constant irreducible polynomials with size and $\bar{\alpha}(f, y)$ bounded:

Corollary 4.3.4. Let C, s, y be positive reals with $C \geq 1$; and d, m, N be integers with $d, N \geq 1$. Suppose that $p \nmid N$, for all primes $p \leq y$. Then, for any real number $l \geq 1$, there are at most $2l$ non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\bar{\alpha}(f, y) \leq \sum_{p \leq y} \frac{\log p}{p-1} - \sqrt{\left(\left(1 - \frac{1}{l}\right) 2d \log C + \frac{1}{l} \sum_{p \leq y} p \log p\right) \sum_{p \leq y} \frac{p^3 \log p}{(p^2-1)^2}}.$$

Proof. The corollary is obtained from Theorem 4.3.2 by setting

$$\{(\mathfrak{p}_1, \beta_1), \ldots, (\mathfrak{p}_n, \beta_n)\} = \left\{ \left(\mathfrak{p}_{p,r}, \frac{p \log p}{p^2 - 1} \right) \mid (p, r) \in \mathcal{U} \text{ and } p \leq y \right\},$$

and noting that

$$\bar{\alpha}(f, y) = \sum_{p \leq y} \frac{\log p}{p - 1} - \sum_{i=1}^{n} \sigma(f, \mathfrak{p}_i)\beta_i. \qquad \square$$

Example 4.3.5. Table 4.1 contains examples of the bounds obtained from Corollary 4.3.4. Let m and N be integers such that N is nonzero and free of prime divisors less than 10000. Then a nonempty entry in Table 4.1a corresponding to C and y contains an upper bounds on the number of non-constant irreducible polynomials $f \in \langle N, x - m \rangle$ with $\deg f \leq 3$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{6}}$ and $\bar{\alpha}(f, y) \leq 0$. Similarly, the entries of Tables 4.1b, 4.1c and 4.1d contain upper bounds on the number of all such polynomials with $\bar{\alpha}(f, y) \leq \varepsilon$, for $\varepsilon = 1$, -1 and -2, respectively. In each table, an entry containing a dash corresponds to a value of C and y for which Corollary 4.3.4 does not apply.

Table 4.1: Bounds for Example 4.3.5.

Table 4.1a: $\bar{\alpha}(f, y) \leq 0$

C	$y = 100$	$y = 1000$	$y = 10000$
1	1899	144167	11066636
2	26995	378128	20574633
3	-	7461248	41362338
4	-	-	146083942

Table 4.1b: $\bar{\alpha}(f, y) \leq -1$

C	$y = 100$	$y = 1000$	$y = 10000$
1	1229	107670	8891047
2	3084	200167	14141365
3	26503	402373	21604070
4	-	1420518	34534713
5	-	-	64461042
6	-	-	220779227

Table 4.1c: $\bar{\alpha}(f, y) \leq -1.5$

C	$y = 100$	$y = 1000$	$y = 10000$
1	1020	94414	8036340
2	2035	158733	12095321
3	4883	263899	17167487
4	741073	497996	24438859
5	-	1596512	36396308
6	-	-	60637371
7	-	-	138796736

Table 4.1d: $\bar{\alpha}(f, y) \leq -2$

C	$y = 100$	$y = 1000$	$y = 10000$
1	860	83463	7299206
2	1484	130046	10499454
3	2581	193086	14121084
4	5434	294311	18696869
5	38188	496011	24973925
6	-	1127183	34414014
7	-	-	50578542
8	-	-	85275302
9	-	-	215937570

To end the section, a final corollary to Theorem 4.3.2 is now given. The corollary provides a sufficient condition under which the approach to polynomial generation described in Section 4.1 is combinatorially feasible. This condition is used to evaluate the performance of the polynomial generation algorithm developed in the next section.

Corollary 4.3.6. Let C, s be positive reals with $C > 1$; and d, m, N be integers with $d, N \geq 1$. Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$; and $\sum_{i=1}^n \log p_i > 2d \log C$. Given positive real weights $z_1, \ldots, z_n$ and any tolerance parameter $\varepsilon > 0$, there are at most polynomially many (in $1/\varepsilon$ and $\sum_{i=1}^n \log p_i$) non-constant irreducible polynomials $f \in \langle N, x - m \rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\sum_{i=1}^{n} \sigma(f, \mathfrak{p}_i) z_i \log p_i \geq \sqrt{2d \log C \left(\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon z_{\max}^2 \right)}, \tag{4.28}$$

where $z_{\max} = \max_{1 \leq i \leq n} z_i$.

Proof. It is sufficient to show that there are at most $2l$ polynomials that satisfy the conditions of the corollary, where

$$l = \frac{1}{\varepsilon z_{\max}^2} \left(\frac{\sum_{i=1}^n \log p_i}{2d \log C} - 1 \right) \sum_{i=1}^{n} z_i^2 \log p_i > 0.$$

If $l < 1$, then an application of the Cauchy-Schwarz inequality shows that the right hand side of (4.28) is greater than $\sum_{i=1}^n z_i \log p_i$, and thus (4.28) is never satisfied:

$$\sqrt{2d \log C \left(\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon z_{\max}^2 \right)} > \sqrt{\left(\sum_{i=1}^{n} \log p_i \right) \left(\sum_{i=1}^{n} z_i^2 \log p_i \right)} \geq \sum_{i=1}^{n} z_i \log p_i.$$

Therefore, assume that $l \geq 1$. Then applying Theorem 4.3.2 with $\beta_i = z_i \log p_i$, for $1 \leq i \leq n$, shows that there are at most $2l$ polynomials that satisfy the conditions of the corollary. □

4.4 An Initial Algorithm

An integer polynomial h may be factored over $\mathbb{Q}$ in time polynomial in $\deg h$ and $\log \|h\|_2$ using existing algorithms [103, 16]. Therefore, the problem of developing an algorithm based on the approach to polynomial generation introduced in Section 4.1 reduces to that of determining an efficient method for constructing a nonzero polynomial $h \in \langle N, x - m \rangle^{z_0} \cdot \prod_{i=1}^n \mathfrak{p}_i^{z_i}$ such that $\deg h$ and $\|h\|_{2,s}$ are small. Such a method has already been developed by Guruswami, Sahai and Sudan [69] as part of their weighted list decoding algorithm for Chinese remainder codes. Here, their method is used to develop an initial realisation of the approach described in Section 4.1. In Section 4.4.1, parameter selection for the algorithm is considered. There it is found that the algorithm's complexity is too large to justify its practical application. In Section 4.5, possible improvements to the algorithm and generalisations of the approach of Section 4.1 are discussed. To begin this section, the method used to construct a suitable polynomial h in the decoding algorithm of Guruswami et al. is briefly reviewed in the context of polynomial generation.

Guruswami, Sahai and Sudan observe that the polynomials of degree *at most* l in $\langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ may be viewed as an integer lattice $L \subset \mathbb{Z}^{l+1}$. Therefore, by appropriately scaling L, the problem of finding a polynomial $h \in \langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ with $\deg h \leq l$ and $\|h\|_{2,s}$ small, is reduced to that of finding a short vector in a lattice. Provided a basis for L can be computed efficiently, lattice reduction may then be used to find a short vector in the scaled lattice. Guruswami et al. provide such a method based on the following lemma [69, Lemma 2]:

Lemma 4.4.1. Let $q, r, z, l \in \mathbb{Z}$ with $q \neq 0$ and $z, l > 0$. Suppose that $f \in \langle q, x-r\rangle^{z}$ and $\deg f \leq l$. Then f can be expressed as an integer combination of the polynomials $q^{z-j}\,(x-r)^j$, for $0 \leq j \leq \min\{z, l\}$; and, if $l > z$, the additional polynomials $x^j(x-r)^z$, for $1 \leq j \leq l-z$.

Lemma 4.4.1 provides a basis for the lattice L_{0,z_0}, corresponding to the polynomials of degree at most l in $\langle N, x-m\rangle^{z_0}$; and a basis for the lattice L_{i,z_i}, corresponding to the polynomials of degree at most l in $\mathfrak{p}_i^{z_i}$, for $1 \leq i \leq n$. These bases are then used to compute a basis for the lattice $L = \bigcap_{i=0}^{n} L_{i,z_i}$ by repeatedly applying the method described by Guruswami et al. [69, Appendix B] for computing a basis for the intersection of two lattices.

Using the approach just described, the following algorithm is obtained:

Algorithm 4.4.2.

Input: Nonzero integers m and N, with $0 \leq m < N$, and pairwise comaximal ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$. Positive integers s, $z_0, \ldots, z_n$ and l.

Output: A list of integer polynomials.

0. For $1 \leq i \leq n$, define the following families of polynomials in $\mathbb{Z}[x]$:

$$\begin{aligned} b_{i,j}(x) &= p_i^{z_i - j}\,(x-r_i)^j, &&\text{for } 0 \leq j \leq \min\{z_i, l\}; \text{ and}\\ b_{i,j}(x) &= x^{j-z_i}(x-r_i)^{z_i}, &&\text{for } z_i + 1 \leq j \leq l. \end{aligned}$$

Similarly, define polynomials

$$\begin{aligned} b_{0,j}(x) &= N^{z_0 - j}\,(x-m)^j, &&\text{for } 0 \leq j \leq \min\{z_0, l\}; \text{ and}\\ b_{0,j}(x) &= x^{j-z_0}(x-m)^{z_0}, &&\text{for } z_0 + 1 \leq j \leq l. \end{aligned}$$

1. Let δ_l be the map that sends an integer polynomial $\sum_{i=0}^{l} a_i x^i$ of degree at most l to the vector $(a_0, \ldots, a_l) \in \mathbb{Z}^{l+1}$. Compute the vectors $\boldsymbol{b}_{i,j} = \delta_l(b_{i,j})$, for $0 \leq i \leq n$, $0 \leq j \leq l$.

2. For $0 \leq i \leq n$, let $L_{i,z_i} \in \mathbb{Z}^{l+1}$ be the lattice generated by the vectors $\boldsymbol{b}_{i,0} \ldots, \boldsymbol{b}_{i,l}$. By repeatedly applying the method described by Guruswami et al. [69, Appendix B], compute a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{l+1})$ for the intersection lattice $L = \bigcap_{i=0}^{n} L_{i,z_i}$.

3. Let $S = \operatorname{diag}(1, s, \ldots, s^l)$. Compute an LLL-reduced basis $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{l+1})$ for the lattice L_S.

4. Use the polynomial time algorithm of Lenstra, Lenstra and Lovás [103] to find all non-constant irreducible factors of $h = \delta_l^{-1}(\boldsymbol{v}_1 S^{-1})$ in $\mathbb{Z}[x]$.

5. Return all factors found in Step 4.

The following theorem provides a condition under which a polynomial is returned by Algorithm 4.4.2:

Theorem 4.4.3. *Amongst the polynomials returned by Algorithm 4.4.2 are all non-constant irreducible polynomials $f \in \mathbb{Z}[x]$ such that*

$$N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} > \|f\|_{2,s}^{l} \cdot \left[2^{\frac{l}{4}} \cdot \left(N^{\binom{z_0+1}{2}} \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{l+1}}\right]^{\deg f}. \tag{4.29}$$

Moreover, the algorithm runs in time polynomial in n, l, $\log s$, $z_0, \ldots, z_n$, $\log p_1, \ldots, \log p_n$ and $\log N$.

Proof. For each i, $0 \le i \le n$, it is readily verified that the degrees of the $l+1$ polynomials $b_{i,j}$ are all distinct and less than $l+1$. Therefore, it is possible to construct an $(l+1)\times(l+1)$ lower triangular matrix B_i with row vectors $\boldsymbol{b}_{i,0} \ldots, \boldsymbol{b}_{i,l}$ such the diagonal elements are precisely the leading coefficients of the polynomials $b_{i,0}, \ldots, b_{i,l}$. Hence, B_i is a basis matrix for the lattice L_{i,z_i}, for $0 \le i \le n$. Consequently, L_{i,z_i} is a full-rank sublattice of $\mathbb{Z}^{l+1}$ and

$$\det L_{i,z_i} = |\det B_i| = \prod_{j=0}^{\min\{z_i,l\}} p_i^{z_i-j} \le p_i^{\binom{z_i+1}{2}}, \quad \text{for } 1 \le i \le n.$$

Similarly, L_{0,z_0} is a full-rank sublattice of $\mathbb{Z}^{l+1}$ and $\det L_{0,z_0} = N^{\binom{z_0+1}{2}}$. Since $L = \bigcap_{i=0}^{n} L_{i,z_i}$ and $[\mathbb{Z}^{l+1} : L_{i,z_i}]$ is finite for $0 \le i \le n$, it follows that

$$\left[\mathbb{Z}^{l+1} : L\right] \le \prod_{i=0}^{n} \left[\mathbb{Z}^{l+1} : L_{i,z_i}\right] = \prod_{i=0}^{n} \det L_{i,z_i} = N^{\binom{z_0+1}{2}} \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}.$$

Hence, L is a full-rank sublattice of $\mathbb{Z}^{l+1}$. Thus,

$$\det L_S = |\det S| \cdot \det L = |\det S| \cdot [\mathbb{Z}^{l+1} : L] \cdot \det \mathbb{Z}^{l+1} = s^{\binom{l+1}{2}} \cdot [\mathbb{Z}^{l+1} : L].$$

The vector $\boldsymbol{v}_1$ is nonzero and Theorem 3.1.2 implies that $\|\boldsymbol{v}_1\| \le 2^{l/4} \det(L_S)^{1/(l+1)}$. Hence, h is nonzero and

$$\|h\|_{2,s} = s^{-\frac{\deg h}{2}} \cdot \|\boldsymbol{v}_1\|_2 \le s^{\frac{l-\deg h}{2}} \cdot 2^{\frac{l}{4}} \cdot \left(N^{\binom{z_0+1}{2}} \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{l+1}}. \tag{4.30}$$

By construction, h is contained in the intersection of the ideals $\mathfrak{p}_1^{z_1}, \ldots, \mathfrak{p}_n^{z_n}$ and $\langle N, x-m\rangle^{z_0}$. However, these ideals are pairwise comaximal, since $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ are pairwise comaximal and $p_i \nmid N$, for $1 \le i \le n$.

Therefore, $h \in \langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$.

If no polynomial satisfies the conditions of the theorem, then its first assertion is vacuously true. Therefore, assume there exists a polynomial $f \in \mathbb{Z}[x]$ that satisfies the conditions of the theorem. Then

$$\begin{aligned} \|f\|_{2,s}^{\deg h} \cdot \|h\|_{2,s}^{\deg f} &\leq \left(\frac{\|f\|_{2,s}}{s^{\frac{\deg f}{2}}}\right)^{\deg h - l} \cdot \|f\|_{2,s}^{l} \cdot \left[2^{\frac{l}{4}} \cdot \left(N^{\binom{z_0+1}{2}} \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{l+1}}\right]^{\deg f} \\ &< N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i}, \end{aligned}$$

since $\|f\|_{2,s} \geq s^{\frac{\deg f}{2}}$. Moreover, Lemma 4.1.1 implies that $N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i}$ divides $\operatorname{Res}(f,h)$. As a result, if h is a constant polynomial, then

$$N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \leq |h|^{\deg f} \leq \|f\|_{2,s}^{\deg h} \cdot \|h\|_{2,s}^{\deg f} < N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i},$$

since $\|f\|_{2,s} \geq 1$, which is absurd. Therefore, h is non-constant and Lemma 4.1.2 implies that $\operatorname{Res}(f,h) = 0$. Consequently, the irreducibility of f and (4.6) imply that f divides h over $\mathbb{Q}$. Hence, f is returned by Algorithm 4.4.2, thus proving the first assertion of the theorem.

Steps 1 and 2 of Algorithm 4.4.2 require time polynomial in n, l $z_0, \ldots, z_n$, $\log p_1, \ldots, \log p_n$ and $\log N$. Steps 3 and 4 require time polynomial n, l, $\log s$, $z_0, \ldots, z_n$, $\log p_1, \ldots, \log p_n$ and $\log N$, with the time bound for Step 4 following from (4.30). Algorithm 4.4.2 therefore runs in the stated time. □

4.4.1 Parameter Selection for Algorithm 4.4.2

In this section, parameter selection for Algorithm 4.4.2 is used to develop an algorithm for generating number field sieve polynomials with specified size and root properties. To begin, polynomial generation with arbitrary (positive) real weights $z_1, \ldots, z_n$ is considered through the careful selection of parameters for Algorithm 4.4.2. Then the performance of the resulting algorithm is evaluated against the theoretical bounds obtained in Section 4.3. Finally, polynomial generation under appropriate choices of weights is considered.

Theorem 4.4.4. *Let $C \geq 1$ be a real number; d, s, N be positive integers; m be an integer such that $0 \leq m < N$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i,r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. Given positive real weights $z_1, \ldots, z_n$ and any tolerance parameter $\varepsilon > 0$, there exists an algorithm that returns all non-constant irreducible polynomials $f \in \langle N, x-m\rangle$ with $\deg f \leq d$ and*

$\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ *such that*

$$\sum_{i=1}^{n} \sigma^*(f, \mathfrak{p}_i) z_i \log p_i \geq \sqrt{2d \log\left(2^{\frac{d}{4}} C\right) \left(\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon z_{\max}^2\right)}, \tag{4.31}$$

where $z_{\max} = \max_{1 \leq i \leq n} z_i$. *Moreover, the algorithm runs in time polynomial in* n, d, $\log s$, $\log C$, $\sum_{i=1}^{n} \log p_i$, $\log N$ *and* $1/\varepsilon$.

The proof of Theorem 4.4.4 presented here adapts arguments of Guruswami, Sahai, and Sudan [69, Theorem 4] (see also [68, Theorem 7.12]).

Proof. The condition (4.31) is invariant under scaling of the parameters $z_1, \ldots, z_n$. Thus, assume without loss of generality that $z_{\max} \leq 1$. Set $z_i^* = \lceil A z_i \rceil$, for $1 \leq i \leq n$, where A is a positive real parameter to be specified later in the proof. Additionally, let z_0^* and l be positive integers parameters to be specified later. Consider following algorithm: first, apply Algorithm 4.4.2 with parameters m, N, $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$, s, $z_0^*, \ldots, z_n^*$ and l; second, return only those polynomials $f \in \mathbb{Z}[x]$ from the output of Algorithm 4.4.2 that satisfy $f \in \langle N, x - m \rangle$, $\deg f \leq d$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ and (4.31). By specifying the parameters z_0^*, l and A it is shown that this algorithm works and satisfies the condition of the theorem.

Theorem 4.4.3 implies that Algorithm 4.4.2, when applied with parameters m, N, $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$, s, $z_0^*, \ldots, z_n^*$ and l, has amongst its outputs all polynomials $f \in \langle N, x - m \rangle$ such that $\deg f \leq d$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ and

$$\prod_{i=1}^{n} p_i^{\sigma^*(f, \mathfrak{p}_i) z_i^*} > 2^{\frac{dl}{4}} \cdot C^l \cdot N^{\frac{l}{2d} + \frac{d}{l+1}\binom{z_0^*+1}{2} - z_0^*} \cdot \left(\prod_{i=1}^{n} p_i^{\binom{z_i^*+1}{2}}\right)^{\frac{d}{l+1}}. \tag{4.32}$$

Moreover, it follows from Theorem 4.4.3 that the algorithm described above runs in time polynomial in n, l, $\log s$, $\log C$, z_0^*, A, $\log p_1, \ldots, \log p_n$ and $\log N$. The proof that the algorithm works and satisfies the condition of the theorem is completed by specifying the parameters z_0^*, l and A that are polynomial in d, $\log C$, $\sum_{i=1}^{n} \log p_i$, $\log N$ and $1/\varepsilon$, and for which (4.32) is satisfied by all polynomials $f \in \mathbb{Z}[x]$ that satisfy (4.31).

Set $z_0^* = \left\lfloor \frac{1}{d}(l+1) \right\rceil$, i.e., the integer nearest to the (real) value of z_0^* that minimises the exponent of N in (4.32), and impose the additional requirement that $l \geq d$, so that z_0^* is positive. Then

$$\frac{l+1}{d} - \frac{1}{2} < z_0^* \leq \frac{l+1}{d} \quad \text{or} \quad \frac{l+1}{d} - 1 < z_0^* \leq \frac{l+1}{d} - \frac{1}{2},$$

and, in either case, the exponent of N in (4.32) is at most $(2d-1)/2d$. Hence, for $l \geq d$, the condition

(4.32) is satisfied whenever

$$\prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i^*} > 2^{\frac{dl}{4}} \cdot C^l \cdot N^{1-\frac{1}{2d}} \cdot \left(\prod_{i=1}^{n} p_i^{\binom{z_i^*+1}{2}} \right)^{\frac{d}{l+1}}. \tag{4.33}$$

The inequalities $Az_i \leq z_i^* < Az_i + 1$, for $1 \leq i \leq n$, imply that (4.33) is satisfied whenever

$$\sum_{i=1}^{n} \sigma^*(f,\mathfrak{p}_i) z_i \log p_i \geq \frac{l}{A} \log\left(2^{\frac{d}{4}}C\right) + \frac{1}{A} \log N^{1-\frac{1}{2d}} + \frac{dA}{2(l+1)} \sum_{i=1}^{n} \left(z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2} \right) \log p_i. \tag{4.34}$$

Define $Z_i = z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2}$, for $1 \leq i \leq n$, and set

$$l = \left\lceil A \sqrt{\frac{d \sum_{i=1}^{n} Z_i \log p_i}{2 \log(2^{\frac{d}{4}}C)}} \right\rceil - 1.$$

Then (4.34) is satisfied whenever

$$\sum_{i=1}^{n} \sigma^*(f,\mathfrak{p}_i) z_i \log p_i \geq \frac{1}{A} \log N^{1-\frac{1}{2d}} + \sqrt{2d \log\left(2^{\frac{d}{4}}C\right) \cdot \sum_{i=1}^{n} Z_i \log p_i}. \tag{4.35}$$

If A is chosen to satisfy

$$A \geq \max\left\{ \frac{10 z_{\max}}{\varepsilon} \sum_{i=1}^{n} \log p_i, \frac{1}{B} \log N^{1-\frac{1}{2d}}, (d+1)\sqrt{\frac{2\log(2^{\frac{d}{4}}C)}{d\sum_{i=1}^{n} Z_i \log p_i}} \right\},$$

for some positive constant B, then $l \geq d$ and

$$\frac{1}{A} \log N^{1-\frac{1}{2d}} + \sqrt{2d \log\left(2^{\frac{d}{4}}C\right) \cdot \sum_{i=1}^{n} Z_i \log p_i} \leq B + \sqrt{2d \log\left(2^{\frac{d}{4}}C\right) \left(\sum_{i=1}^{n} z_i^2 \log p_i + \frac{\varepsilon}{2} \right)}.$$

Furthermore, if

$$B \leq \frac{\varepsilon}{4} \cdot \sqrt{\frac{2d\log(2^{\frac{d}{4}}C)}{\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon}},$$

then

$$B + \sqrt{2d \log\left(2^{\frac{d}{4}}C\right) \left(\sum_{i=1}^{n} z_i^2 \log p_i + \frac{\varepsilon}{2} \right)} \leq \sqrt{2d \log\left(2^{\frac{d}{4}}C\right) \left(\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon \right)}.$$

(The upper bound on B is obtained by using the mean value theorem to bound the difference of the two roots occurring in this inequality.) Therefore, set

$$A = \max\left\{ \frac{10 z_{\max}}{\varepsilon} \sum_{i=1}^{n} \log p_i, \frac{4}{\varepsilon} \log N^{1-\frac{1}{2d}} \sqrt{\frac{\sum_{i=1}^{n} z_i^2 \log p_i + \varepsilon}{2d\log(2^{\frac{d}{4}}C)}}, (d+1)\sqrt{\frac{2\log(2^{\frac{d}{4}}C)}{d\sum_{i=1}^{n} z_i^2 \log p_i}} \right\}.$$

Then $l \geq d$ and (4.35) holds for all $f \in \mathbb{Z}[x]$ that satisfy (4.31). Moreover, for this choice of A, the parameters z_0^*, l and A are all polynomial in d, $\log C$, $\sum_{i=1}^{n} \log p_i$, $\log N$ and $1/\varepsilon$. □

Remark 4.4.5. It is not clear under which choices of parameters the conditions on the polynomials returned by the algorithm described in Theorem 4.4.4 are satisfiable. If the conditions are not satisfiable, then the algorithm proves it in polynomial time, and this information may be used to narrow the search space for polynomials. However, there is also potential for computational effort to be wasted by having to repeatedly adjust parameters and re-execute the algorithm so that polynomials are found. Therefore, it is of theoretical and practical interest to determine under which choices of parameters the conditions in Theorem 4.4.4 are satisfiable.

The performance of the algorithm described in Theorem 4.4.4 is now evaluated by comparing against Corollary 4.3.6. At first glance, the bounds (4.31) and (4.28) appear to be incomparable, since the left hand sides of the two inequalities are not necessary equal, due to the appearance of both σ^* and σ. However, the following lemma allows parameters to be constructed for which the two bounds are comparable:

Lemma 4.4.6. Let $f \in \mathbb{Z}[x]$ be primitive and $(p, r) \in \mathcal{U}$. Then $p^{\sigma^*(f,\mathfrak{p}_{p,r})(\sigma^*(f,\mathfrak{p}_{p,r})-1)}$ divides $\operatorname{Res}(f, f')$.

Proof. Let $f \in \mathbb{Z}[x]$ be primitive and $z = \sigma^*(f, \mathfrak{p}_{p,r})$, where $(p, r) \in \mathcal{U}$. If $z = 0$ or 1, then $p^{z(z-1)}$ trivially divides $\operatorname{Res}(f, f')$. Therefore, assume that $z \geq 2$.

Lemma 4.4.1 implies that f can be written as an integer linear combination of the polynomials

$$b_i = p^{z-i}\,(x-r)^i, \text{ for } 0 \leq i \leq z; \quad \text{and} \quad b_i = x^{i-z}(x-r)^z, \text{ for } i > z.$$

Thus $\deg f \geq z$, otherwise f is not primitive. Furthermore, f' is an integer linear combination of the polynomials b_i', for $i \geq 1$. By computing derivatives, it is readily verified that $\sigma^*(b_i', \mathfrak{p}_{p,r}) \geq z - 1$, for all $i \geq 1$. Therefore, $\sigma^*(f', \mathfrak{p}_{p,r}) \geq z - 1$, where $z - 1 \geq 1$ by assumption. Hence, Lemma 4.1.1 implies that $p^{z(z-1)}$ divides $\operatorname{Res}(f, f')$. □

For any non-constant irreducible polynomial $f \in \mathbb{Z}[x]$ with $\sigma^*(f, \mathfrak{p}_{p,r}) > 1$, property (4.6) and Lemma 4.4.6 imply that $p \leq \sqrt{|\operatorname{Res}(f, f')|}$. Additionally, for any polynomial $f \in \mathbb{Z}[x]$ that satisfies the size and degree constraints of Theorem 4.4.4,

$$|\operatorname{Res}(f, f')| \leq \|f\|_{2,s}^{d-1} \cdot \|f'\|_{2,s}^{d} \leq d^d s^{-\frac{d}{2}} \|f\|_{2,s}^{2d-1} \leq d^d s^{-\frac{d}{2}} C^{2d-1} N^{1-\frac{1}{2d}}.$$

Hence, for any choice of parameters d, s, C, m, N and ε, it is possible to select ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ and their corresponding weights $z_1, \ldots, z_n$ such that, for any non-constant irreducible polynomials $f \in \mathbb{Z}[x]$ that satisfies the corresponding size and degree constraints of Theorem 4.4.4, the difference between $\sum_{i=1}^{n} \sigma^*(f, \mathfrak{p}_i) z_i \log p_i$ and $\sum_{i=1}^{n} \sigma(f, \mathfrak{p}_i) z_i \log p_i$ is arbitrarily small. Comparing conditions (4.31) and (4.28) for these parameters therefore suggests that the algorithm described in Theorem 4.4.4 does not perform optimally.

The following alternative to Theorem 4.4.4 considers polynomial generation with Algorithm 4.4.2 when the parameter l, which determines the dimension of the lattices in the algorithm, is fixed:

Theorem 4.4.7. *Let $C \geq 1$ be a real number; d, s, N be positive integers; m be an integer such that $0 \leq m < N$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. Given positive real weights $z_1, \ldots, z_n$ and any integer $l \geq d$, there exists an algorithm that returns all non-constant irreducible polynomials $f \in \langle N, x - m \rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that*

$$\sum_{i=1}^{n} \sigma^*(f, \mathfrak{p}_i) z_i \log p_i \geq \sqrt{\frac{2d}{l+1}\left(l \log\left(2^{\frac{d}{4}} C\right) + \log N^{1-\frac{1}{2d}} + \frac{d}{l+1}\sum_{i=1}^{n} \log p_i\right) \sum_{i=1}^{n} z_i^2 \log p_i} + \frac{3d}{2(l+1)} \sum_{i=1}^{n} z_i \log p_i. \quad (4.36)$$

Moreover, the algorithm runs in time polynomial in n, d, $\log s$, $\log C$, $\sum_{i=1}^{n} \log p_i$, $\log N$ and l.

Proof. The proof follows that of Theorem 4.4.4, apart from the selection of the parameters A and l. Here, the parameter l is provided and, accordingly, does not require selection. The condition $l \geq d$ implies that $z_0^* = \left\lfloor \frac{1}{d}(l+1) \right\rfloor$ is nonzero, as required. The bound (4.36) is obtained by choosing the parameter A to minimise the right hand side of (4.34). That is, set $A = \sqrt{X/Y}$, where

$$X = l \log\left(2^{\frac{d}{4}} C\right) + \log N^{1-\frac{1}{2d}} + \frac{d}{l+1}\sum_{i=1}^{n} \log p_i \quad \text{and} \quad Y = \frac{d}{2(l+1)} \sum_{i=1}^{n} z_i^2 \log p_i.$$

Then the right hand side of (4.34) becomes

$$\frac{1}{A} X + AY + \frac{3d}{2(l+1)} \sum_{i=1}^{n} z_i \log p_i = 2\sqrt{XY} + \frac{3d}{2(l+1)} \sum_{i=1}^{n} z_i \log p_i.$$

By substituting in X and Y, this is seen to equal the right hand side of (4.36). Finally, for this choice of A, the parameters $z_0^*, \ldots, z_n^*$ are all polynomial in d, $\log C$, $\sum_{i=1}^{n} \log p_i$, $\log N$ and l. □

To end this section, polynomial generation under two common choices of weights is considered: first, the contributions of roots modulo distinct primes are weighted uniformly; secondly, weights are chosen to bound $\bar{\alpha}(f, y)$. For the latter choice of weights, an example of parameter selection is then provided.

Corollary 4.4.8. Let $C \geq 1$ be a real number; d, s, N be positive integers; and m be an integer such that $0 \leq m < N$. Given positive real numbers y and ε such that all prime factors of N exceed y, there exists an algorithm that returns all non-constant irreducible polynomials $f \in \langle N, x - m \rangle$ with

$\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\sum_{p \leq y} \sigma^*(f,p) \geq \sqrt{2d \log\left(2^{\frac{d}{4}} C\right) \left(\sum_{p \leq y} \frac{p}{\log p} + \varepsilon\right)},$$

where the sums are over all primes $p \leq y$. Moreover, the algorithm runs in time polynomial in d, $\log s$, $\log C$, $\log N$, $\sum_{p \leq y} p$ and $1/\varepsilon$.

Proof. Apply Theorem 4.4.4 with

$$\{(\mathfrak{p}_1, z_1), \ldots, (\mathfrak{p}_n, z_n)\} = \left\{\left(\mathfrak{p}_{p,r}, \frac{1}{\log p}\right) \mid (p,r) \in \mathcal{U} \text{ and } p \leq y\right\}.$$

For this choice of parameters, $n = \sum_{p \leq y} p$, leading to the stated running time. □

Corollary 4.4.9. Let $C \geq 1$ be a real number; d, s, N be positive integers; and m be an integer such that $0 \leq m < N$. Given positive real numbers y and ε such that all prime factors of N exceed y, there exists an algorithm that returns all non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\bar{\alpha}(f,y) \leq \sum_{p \leq y} \frac{\log p}{p-1} - \sqrt{2d \log\left(2^{\frac{d}{4}} C\right) \left(\sum_{p \leq y} \frac{p^3 \log p}{(p^2-1)^2} + \varepsilon\right)}, \tag{4.37}$$

where the sums are over all primes $p \leq y$. Moreover, the algorithm runs in time polynomial in d, $\log s$, $\log C$, $\log N$, $\sum_{p \leq y} p$ and $1/\varepsilon$.

Proof. By applying Theorem 4.4.4 with

$$\{(\mathfrak{p}_1, z_1), \ldots, (\mathfrak{p}_n, z_n)\} = \left\{\left(\mathfrak{p}_{p,r}, \frac{p}{p^2-1}\right) \mid (p,r) \in \mathcal{U} \text{ and } p \leq y\right\}, \tag{4.38}$$

it follows that there exists an algorithm that returns all irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\sum_{p \leq y} \frac{\log p}{p-1} - \sum_{i=1}^{n} \sigma^*(f, \mathfrak{p}_i) \frac{p_i \log p_i}{p_i^2 - 1} \leq \sum_{p \leq y} \frac{\log p}{p-1} - \sqrt{2d \log\left(2^{\frac{d}{4}} C\right) \left(\sum_{p \leq y} \frac{p^3 \log p}{(p^2-1)^2} + \varepsilon\right)}. \tag{4.39}$$

Moreover, the algorithm runs in time polynomial in d, $\log s$, $\log C$, $\log N$, $\sum_{p \leq y} p$ and $1/\varepsilon$. The corollary then follows from the observation that the left hand side of (4.39) is less than or equal to $\bar{\alpha}(f,y)$, for all $f \in \mathbb{Z}[x]$. □

Example 4.4.10. Let $N = 10^{170} + 7$. This choice of N is free of prime factors less than 10000 and is representative of a mid-sized value suitable for factorisation by a pair of cubic polynomials. In this example, parameter selection for Algorithm 4.4.2 is considered with the aim of producing cubic polynomials $f \in \mathbb{Z}[x]$ with $\|f\|_{2,s} \leq C \cdot N^{1/6}$, for some $C > 0$, and $\bar{\alpha}(f, y) \leq -2$. Accordingly, set $d = 3$ and let parameters $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ and $z_1, \ldots, z_n$ be chosen according to (4.38), for some $y \geq 2$. In the parameter selections of both Theorem 4.4.4 and Theorem 4.4.7, the parameter s affects neither the bounds obtained nor the choice of the remaining parameters. As a result, its selection is not considered in this example. Moreover, it is assumed that a brute-force search is performed over the parameter m. Therefore, only the parameters C, y and, depending on whether parameters are chosen according to Theorem 4.4.4 or Theorem 4.4.7, the parameter ε or l remain to be chosen.

Table 4.2 contains example of parameters (n, C, A and l) obtained from the proof of Theorem 4.4.4. For each value of y, the tolerance parameter $\varepsilon = 2/3$ is used and the parameter C is chosen so that the right hand side of (4.37) is equal to -2, i.e., C is taken as large as possible. The values in Table 4.2 show that C is restricted to extremely small values. For such small values of C, it is unlikely that a pair of cubics with nonzero resultant exists for most values of m.

Table 4.2: Parameters for Example 4.4.10

			Theorem 4.4.4		Theorem 4.4.7	
y	n	C	A	l	A	l
10	17	1.78	607.2	937	617.7	808
20	77	1.99	845.2	1496	726.5	1142
30	129	2.06	1599.8	2918	774.1	1275
40	197	2.12	2666.7	4977	820.5	1399
50	328	2.20	4866.5	9324	890.3	1578
100	1060	2.41	18886.5	38102	1128.7	2157
1000	76127	3.39	2153390.2	4905092	5505.0	12411

From Table 4.2, it is clear that larger values of C can be achieved by increasing the bound y on the primes considered. However, the remaining parameters in the table exhibit two negative consequences of an increase in y: first, the parameter n increases, thus increasing the number of lattice intersection computations required in Step 2 of Algorithm 4.4.2; and second, the dimension of all lattices in the algorithm (i.e., $l + 1$) also increases. The cost of computing the lattice intersections may be amortised in part by reusing computations from Step 2 (namely, a basis for $\bigcap_{i=1}^{n} L_i$) for many values of m. In contrast, the problem presented by the growth of lattice dimension appears to be insurmountable. A much greater concern is the large lattice dimension that occurs for each value of y. As a brute-force search is performed over the parameter m, the large lattice dimension means that Steps 2–4 of Algorithm 4.4.2 are far too time consuming for the algorithm to be of practical value. For each value of y in Table 4.2, the parameters obtained from the proof of Theorem 4.4.7, with corresponding value of C taken from the table and l chosen so that the right hand side of (4.37) is equal to -2, are significantly smaller (see Table 4.2 once more). However, the improved parameters are once again far too large.

4.4.2 Algorithmic Bounds on Polynomial Selection

Algorithm 4.4.2 is guaranteed to find *all* polynomials with sufficiently small coefficients and good non-projective root properties. Each polynomial found by the algorithm occurs as a factor of the polynomial $h \in \mathbb{Z}[x]$ computed in Step 4. Therefore, the number of degree d polynomials returned by the algorithm is bounded by $(2/d)\deg h$, where the factor of two accounts for units and $\deg h$ is bounded by the parameter l. As a result, Theorem 4.4.3 admits bounds on the existence of number field sieve polynomials with small coefficients and good non-projective root properties. This approach to algorithmically deriving bounds on polynomial is analogous to the use of a list decoding algorithm for Chinese remainder codes by Boneh [25, Section 3.1] to bound the number of smooth integers in short intervals. Here, bounds on polynomial generation parallel to those obtained in Section 4.3 are derived by carefully selecting parameters for Algorithm 4.4.2.

It follows from Minkowski's second theorem (see Section 3.1) that every n-dimensional lattice $\Lambda \subset \mathbb{R}^n$ contains a nonzero vector $\boldsymbol{x}$ satisfying $\|\boldsymbol{x}\|_2 \leq \sqrt{\gamma_n}\det(\Lambda)^{1/n}$, where $\gamma_n \leq 1 + \frac{n}{4}$ is Hermite's constant. By using this fact and modifying arguments from the proofs of Theorem 4.4.3 and Theorem 4.4.4, the following bound on polynomial selection is obtained:

Theorem 4.4.11. *Let C, s be positive reals with $C > 1$; d, m, N be integers with $d, N \geq 1$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$. Given positive real weights $\beta_1, \ldots, \beta_n$ and an integer $l \geq d$, there exist at most $2l/d$ (resp. $2l$) non-constant irreducible polynomials $f \in \langle N, x - m\rangle$ with $\deg f = d$ (resp. $\deg f \leq d$) and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that*

$$\sum_{i=1}^{n} \sigma^*(f, \mathfrak{p}_i)\beta_i \geq \sqrt{\frac{2d}{l+1}\left(l\log C + \frac{d}{l+1}\sum_{i=1}^{n}\log p_i + \log\left(\gamma_{l+1}^{\frac{d}{2}} N^{1-\frac{1}{2d}}\right)\right)\sum_{i=1}^{n}\frac{\beta_i^2}{\log p_i}} + \frac{3d}{2(l+1)}\sum_{i=1}^{n}\beta_i.$$

Proof. For positive integers $z_0, \ldots, z_n$, modifying the proof of Theorem 4.4.3 to use the bound on the shortest vector in the lattice L_S provided by Minkowski's second theorem in place of the bound provided by Theorem 3.1.2 shows that there exists a nonzero polynomial $h \in \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ such that $\deg h \leq l$ and

$$\|h\|_{2,s} \leq s^{\frac{l - \deg h}{2}} \cdot \sqrt{\gamma_{l+1}} \cdot \left(N^{\binom{z_0+1}{2}} \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{l+1}}.$$

As a result, if h is constant and there exists a non-constant irreducible polynomial $f \in \langle N, x - m\rangle$

such that $\deg f \leq d$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ and

$$\prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} > \gamma_{l+1}^{\frac{d}{2}} \cdot C^l \cdot N^{\frac{l}{2d}+\frac{d}{l+1}\binom{z_0+1}{2}-z_0} \cdot \left(\prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}\right)^{\frac{d}{l+1}}, \tag{4.40}$$

then Lemma 4.1.1 implies that

$$N^{z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \leq |h|^{\deg f} \leq \|f\|_{2,s}^{\deg h} \cdot \|h\|_{2,s}^{\deg f} < N^{z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i},$$

which is absurd. Therefore, either h is non-constant or there is no non-constant irreducible polynomial $f \in \langle N, x-m\rangle$ such that $\deg f \leq d$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ and (4.40) holds. If h is non-constant, then Lemma 4.1.2 implies that any non-constant irreducible polynomial $f \in \langle N, x-m\rangle$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ that satisfies (4.40) must divide h over $\mathbb{Q}$. Hence, regardless of the degree of h, all non-constant irreducible $f \in \langle N, x-m\rangle$ such that $\deg f \leq d$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ and (4.40) holds divide h over $\mathbb{Q}$. By considering the maximum number of degree d factors of h, it follows that there can only exist at most $2l/d$ such polynomials with $\deg f = d$. Similarly, by considering the case where h factors completely into linear polynomials, it follows that there exist at most $2l$ such polynomials with $\deg f \leq d$.

Let $A > 0$ be a parameter to be chosen later and set $z_i = \lceil A\beta_i/\log p_i \rceil$, for $1 \leq i \leq n$; and $z_0 = \left\lfloor \frac{1}{d}(l+1) \right\rfloor$, which is nonzero since $l \geq d$. By substituting into (4.40), it follows that, for any $A > 0$, there exist at most $2l/d$ (resp. $2l$) non-constant irreducible polynomials $f \in \langle N, x-m\rangle$ with $\deg f = d$ (resp. $\deg f \leq d$) and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\sum_{i=1}^{n} \sigma^*(f,\mathfrak{p}_i)\beta_i \geq \frac{1}{A}\left(l \log C + \log\left(\gamma_{l+1}^{\frac{d}{2}} N^{1-\frac{1}{2d}}\right)\right) + \frac{dA}{2(l+1)} \sum_{i=1}^{n} \left(\frac{\beta_i^2}{\log p_i} + \frac{3}{A}\beta_i + \frac{2}{A^2}\log p_i\right).$$

The proof is completed by choosing the parameter A such that the right hand side is minimised. □

It is possible to derive corollaries to Theorem 4.4.11 analogous to Corollary 4.3.4 and Corollary 4.3.6 of Theorem 4.3.2. However, details are not provided here. To end this section, examples of the bounds obtained from Theorem 4.4.11 are now provided and compared with the combinatorially derived bounds of Section 4.3:

Example 4.4.12. Table 4.3 contains examples of the bounds obtained from Theorem 4.4.11. Let m be an integer and $N = 10^{170} + 7$. This choice of N is free of prime factors less than 10000 and is representative of a mid-sized value suitable for factorisation by a pair of cubic polynomials. In Tables 4.3a–4.3d, a nonempty entry corresponding to C and y contains an upper bound on the number of irreducible polynomials $f \in \langle N, x-m\rangle$ with $\deg f = 3$, $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{6}}$ and $\bar{\alpha}(f,y) \leq \varepsilon$, for $\varepsilon = 0, -1, -1.5, -2$, respectively. In each table, an entry containing a dash corresponds to a value of C and y for which Theorem 4.4.11 does not apply. Corresponding bounds for the number of polynomials with $\deg f \leq 3$ can be obtained by multiplying each entry in Tables 4.3a–4.3d by three. In all cases, the bounds given in this example are significantly smaller than the combinatorially derived bounds in Example 4.3.5. However, this is a somewhat unfair comparison, since the bounds in Example 4.3.5 hold for all N rather than just $N = 10^{170} + 7$.

Table 4.3: Bounds for Example 4.4.12.

Table 4.3a: $\bar{\alpha}(f, y) \leq 0$

C	$y = 100$	$y = 1000$	$y = 10000$
1	430	1373	10201
2	6044	3196	16669
3	-	57498	30244
4	-	-	97070

Table 4.3b: $\bar{\alpha}(f, y) \leq -1$

C	$y = 100$	$y = 1000$	$y = 10000$
1	300	1167	9132
2	741	1986	13140
3	6350	3749	18650
4	-	12557	27999
5	-	-	49342
6	-	-	159931

Table 4.3c: $\bar{\alpha}(f, y) \leq -1.5$

C	$y = 100$	$y = 1000$	$y = 10000$
1	257	1085	8678
2	507	1690	11944
3	1209	2662	15902
4	184906	4809	21469
5	-	14847	30501
6	-	-	48642
7	-	-	106788

Table 4.3d: $\bar{\alpha}(f, y) \leq -2$

C	$y = 100$	$y = 1000$	$y = 10000$
1	224	1014	8267
2	383	1476	10972
3	662	2093	13952
4	1387	3075	17649
5	9756	5022	22656
6	-	11100	30117
7	-	-	42804
8	-	-	69903
9	-	-	171650

4.5 Future Directions: Improvements and Generalisations

The parameters found in Example 4.4.10 suggest that the approach to polynomial generation introduced in Section 4.1 requires development beyond the initial realisation (Algorithm 4.4.2) provided in Section 4.4. In this section, potential avenues for generalising the approach of Section 4.1 and improving its realisation are discussed. In Section 4.5.1, the approach is modified so that better parameters for Algorithm 4.4.2 may be obtained. In Section 4.5.2, ideas central to the nonlinear algorithms described in Chapter 3 are generalised and incorporated into the initial algorithm. Finally, a multivariate generalisation of the approach of Section 4.1 is introduced in Section 4.5.3 and details related to its realisation are discussed.

4.5.1 Special-$\mathfrak{q}$

In Section 4.4.2, the observation that all polynomials found by Algorithm 4.4.2 occur as a factor of the polynomial h was used to provide bounds on the existence of number field sieve polynomials with small coefficients and good non-projective root properties. However, this observation further imposes a fundamental lower bound on the parameter l in the algorithm: if polynomials $f_1, \ldots, f_t$ are returned by the algorithm, then $\frac{1}{2}\sum_{i=1}^{t} \deg f_i \leq l$. Therefore, it may be necessary to encounter lattices of large dimension when applying Algorithm 4.4.2. In this section, a modification to the algorithm is presented which is aimed reducing the number of polynomials found by the algorithm by imposing greater restrictions on their root properties. The method used is motivated by previous work of Bai, Brent and Thomé [12], Kleinjung [90] and Pollard [142].

Given a pair of number field sieve polynomials $f, g \in \mathbb{Z}[x]$ with a common root m modulo N, define the rotated polynomial $f_{u,v}(x) = f(x) + (ux+v)g(x)$, for all $(u,v) \in \mathbb{Z}^2$. A rotated polynomial $f_{u,v}$ that has few roots modulo small prime powers is less likely to have good root properties (see Section 2.1.2). Motivated by this observation, Bai et al. [12, Section 5] suggested a two-stage method for finding linear rotations, which only performs sieving over pairs $(u,v) \in \mathbb{Z}^2$ such that $f_{u,v}$ is guaranteed to have many roots modulo small primes. In the first stage of their algorithm, Gower's approach [65, Section 4] is followed and the Chinese remainder theorem used to construct an initial pair $(u_0, v_0) \in \mathbb{Z}^2$ such that f_{u_0,v_0} has many roots modulo (very) small prime powers, say $p_1^{e_1}, \ldots, p_t^{e_t} < B$, for some bound B. In the second stage, a modified root sieve is performed modulo primes powers greater than B and restricted to those pairs $(u,v) \in \mathbb{Z}^2$ contained in the subset $\{(u_0 + a\prod_{i=1}^{t} p_i^{e_i}, v_0 + b\prod_{i=1}^{t} p_i^{e_i}) \mid (a,b) \in \mathbb{Z}^2\}$. As a result, only pairs $(u,v) \in \mathbb{Z}^2$ such that $f_{u,v}$ has many roots modulo prime powers $p_1^{e_1}, \ldots, p_t^{e_t}$ are considered by the algorithm. The approach introduced in this section uses Algorithm 4.4.2 to find those polynomials $f \in \langle N, x-m\rangle \cap \mathfrak{q}$ with good size and root properties, where $\mathfrak{q} = \prod_{(p,r)\in\mathcal{Q}} \mathfrak{p}_{p,r}$ for some finite set $\mathcal{Q} \subset \mathcal{U}$. The strategy of Bai et al. is then captured by choosing $\mathcal{Q}$ so that all polynomials returned by the algorithm will have many roots modulo small primes. The ideal $\mathfrak{q}$ is referred to as a *special*-$\mathfrak{q}$ in reference to analogous ideas that occur in linear polynomial generation [90] and lattice sieve [142] algorithms.

Given a special-$\mathfrak{q}$, Algorithm 4.4.2 may be applied to pairwise comaximal ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+t}$, where $\mathfrak{q} = \prod_{i=n+1}^{n+t} \mathfrak{p}_i$. Then the weights $z_{n+1}, \ldots, z_{n+t}$ corresponding to those ideals $\mathfrak{p}_{n+1}, \ldots, \mathfrak{p}_{n+t}$ can be freely chosen. Thus, a careful selection of the weights may be used to leverage an advantage. A similar approach was applied in the proof of Theorem 4.4.4, where the parameter z_0^* corresponding to the ideal $\langle N, x-m\rangle$ was chosen to minimise the contribution of N in (4.32). The influence of utilising a special-$\mathfrak{q}$ in this manner is summarised by the following theorem:

Theorem 4.5.1. *Let $C \geq 1$ be a real number; d, s, N be positive integers; m be an integer such that $0 \leq m < N$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+t} \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i,r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n+t$. Define $\mathfrak{q} = \prod_{i=n+1}^{n+t} \mathfrak{p}_i$, $q = \prod_{i=n+1}^{n+t} p_i$ and suppose that $q^{1/2d} < 2^{\frac{d}{4}}C$. Then, given positive real weights $z_1, \ldots, z_n$ and any integer $l \geq d$, there exists an algorithm that returns all* *non-constant irreducible polynomials* $f \in \langle N, x-m\rangle \cdot \mathfrak{q}$ *with* $\deg f \leq d$ *and* $\|f\|_{2,s} \leq C \cdot N^{1/2d}$ *such that*

$$\sum_{i=1}^{n} \sigma^*(f,\mathfrak{p}_i) z_i \log p_i \geq \sqrt{\frac{2d}{l+1}\left(l \log\left(\frac{2^{\frac{d}{4}}C}{q^{\frac{1}{2d}}}\right) + \left(1-\frac{1}{2d}\right)\log(qN) + \frac{d}{l+1}\sum_{i=1}^{n}\log p_i\right)\sum_{i=1}^{n} z_i^2 \log p_i} + \frac{3d}{2(l+1)}\sum_{i=1}^{n} z_i \log p_i. \quad (4.41)$$

Moreover, the algorithm runs in time polynomial in n, d, $\log s$, $\log C$, $\sum_{i=1}^{n}\log p_i$, $\log q$, $\log N$ *and* l.

The proof follows that of Theorem 4.4.7 with only minor modifications to account for the inclusion of the additional ideals $\mathfrak{p}_{n+1}, \ldots, \mathfrak{p}_{n+t}$.

Proof. The condition (4.41) is invariant under scaling of the parameters $z_1, \ldots, z_n$. Thus, assume without loss of generality that $\max_{1\leq i\leq n} z_i \leq 1$. Let $A > 0$ be a parameter to be determined later and set $z_i^* = \lceil A z_i \rceil$, for $1 \leq i \leq n$. Set $z_i^* = \left\lfloor \frac{1}{d}(l+1)\right\rfloor$, for $n+1 \leq i \leq n+t$; and $z_0^* = \left\lfloor \frac{1}{d}(l+1)\right\rfloor$. Then z_0^* and $z_{n+1}^*, \ldots, z_{n+t}^*$ are all nonzero since $l \geq d$. Therefore, Theorem 4.4.3 implies that Algorithm 4.4.2 can be used to find all non-constant irreducible polynomials $f \in \langle N, x-m\rangle \cdot \mathfrak{q}$ with $\deg f \leq d$ and $\|f\|_{2,s} \leq C \cdot N^{\frac{1}{2d}}$ such that

$$\sum_{i=1}^{n} \sigma^*(f,\mathfrak{p}_i) z_i \log p_i \geq \frac{1}{A}\left(l \log\left(\frac{2^{\frac{d}{4}}C}{q^{\frac{1}{2d}}}\right) + \left(1-\frac{1}{2d}\right)\log(qN)\right) + \frac{dA}{2(l+1)}\sum_{i=1}^{n}\left(z_i^2 + \frac{3}{A}z_i + \frac{2}{A^2}\right)\log p_i,$$

in time polynomial in n, d, $\log s$, $\log C$, A, $\log p_1, \ldots, \log p_n$, $\log q$, $\log N$ and l. The bound (4.41) is obtained by choosing the parameter A to minimise the right hand side of the inequality. Moreover, this choice of A is polynomial in d, $\log C$, $\sum_{i=1}^{n}\log p_i$, $\log q$, $\log N$ and l, leading to the running time in the statement of the theorem. □

A similar result may be obtained for the case where $q^{1/2d} \geq 2^{\frac{d}{4}}C$. However, the following corollary to Lemma 4.1.2 shows that at most one polynomial (up to units) is found whenever $C < q^{1/2d}$:

Corollary 4.5.2. Let $C \geq 1$ be a real number; d, m, N be integers with $d, N \geq 1$; and $\mathfrak{p}_1, \ldots, \mathfrak{p}_{n+t} \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i,r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n+t$. Define $\mathfrak{q} = \prod_{i=n+1}^{n+t}\mathfrak{p}_i$ and $q = \prod_{i=n+1}^{n+t} p_i$. Suppose there exist non-constant irreducible polynomials $f_1, f_2 \in \langle N, x-m\rangle \cdot \mathfrak{q}$ of degree at most d such that $\|f_i\|_{2,s} \leq C \cdot N^{1/2d}$, for $i = 1, 2$, and

$$\sum_{i=1}^{n} \sigma^*(f_1,\mathfrak{p}_{p,r})\sigma^*(f_2,\mathfrak{p}_{p,r}) \log p > 2d \log\left(C q^{-\frac{1}{2d}}\right).$$

Then $f_1 = \pm f_2$.

To end the section, an example of parameter selection in the presence of a special-$\mathfrak{q}$ is now provided and parameters compared with those obtained in Example 4.4.10:

Example 4.5.3. To allow direct comparison with Example 4.4.10, let $N = 10^{170} + 7$ and $d = 3$. For each choice of parameters y and C in Table 4.2, parameter selection for Algorithm 4.4.2 with the aim of producing cubic polynomials $f \in \langle N, x - m\rangle$, where $m \in \mathbb{Z}$, with $\|f\|_{2,s} \leq C \cdot N^{1/6}$ and $\bar{\alpha}(f, y) \leq -2$ is once again considered. However, unlike Example 4.4.10, the polynomials are now required to belong to a special-$\mathfrak{q}$. For this example, the choice $\mathfrak{q} = \mathfrak{p}_{2,0}\,\mathfrak{p}_{2,1}\,\mathfrak{p}_{3,0}$ is used, thus guaranteeing that all polynomials have two roots modulo 2 and at least one root modulo 3. Accordingly, for each choice of y and C, the parameters $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ and $z_1, \ldots, z_n$ are chosen as follows:

$$\{(\mathfrak{p}_1, z_1), \ldots, (\mathfrak{p}_n, z_n)\} = \left\{ \left(\mathfrak{p}_{p,r}, \frac{p}{p^2 - 1} \right) \mid (p, r) \in \mathcal{U} \setminus \{(2,0), (2,1), (3,0)\} \text{ and } p \leq y \right\}.$$

Table 4.4 contains example of parameters (n, A and l) obtained from the proof of Theorem 4.5.1. For each value of y in the table, the corresponding value of l is the least integer such that the right hand side of (4.41) is at most

$$2 - \frac{4}{3}\log 2 - \frac{3}{8}\log 3 + \sum_{p \leq y} \frac{\log p}{p - 1},$$

where the sum is over all primes $p \leq y$. This choice of l then guarantees that all $f \in \langle N, x - m\rangle \cdot \mathfrak{q}$ with $\|f\|_{2,s} \leq C \cdot N^{1/6}$ and $\bar{\alpha}(f, y) \leq -2$ are found. For each value of y in Table 4.4, the corresponding values of A and l are significantly smaller than those provided in Example 4.4.10.

Table 4.4: Parameters for Example 4.5.3

y	n	C	A	l
10	14	1.78	563.5	607
20	74	1.99	609.0	901
30	126	2.06	613.3	969
40	194	2.12	615.7	1018
50	325	2.20	619.6	1075
100	1057	2.41	642.2	1214
1000	76124	3.39	1441.4	3232

4.5.2 Lattice Construction

The efficacy of an algorithm based on the approach outlined in Section 4.1 is determined by its capacity to construct a polynomial $h \in \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ with small coefficients. In Section 4.4, Algorithm 4.4.2 constructed such a polynomial by using lattice reduction to find a short vector in the lattice L, which contained a coefficient vector for each polynomial of degree at most l in $\langle N, x - m\rangle^{z_0} \cdot$

$\prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$. There the size of the quantity $\det(L_S)^{1/(l+1)}$ played a key role in determining the size of the polynomial found. In this section, a potential method for constructing sublattices $L' \subset L$ such that $\det(L'_S)^{1/\dim L'_S} < \det(L_S)^{1/(l+1)}$ is proposed. Such a sublattice cannot be full-rank. Hence, it is necessary to work with lattices which are not full-rank, thus complicating the resulting the analysis. However, from the analysis of Section 4.4, it is deduced that the potential benefits of improved lattice constructions are twofold: first, lattices of lower dimension may be used; and second, the output requirements on size and root properties may be weakened. Throughout this section, notation from Algorithm 4.4.2 is retained.

In the proof of Theorem 4.4.4, the freedom in the choice of the parameter z_0^*, corresponding to the ideal $\langle N, x - m\rangle$, was used to minimise the contribution of N in (4.32). Motivated by further leveraging the advantage gained from this freedom, attention is limited in this section to constructing sublattices of L that are of the form $L' = L'_{0,z_0} \cap L_{1,z_1} \cap \ldots \cap L_{n,z_n}$, for some sublattice $L'_{0,z_0} \subset L_{0,z_0}$. The following lemma provides an upper bound on the determinant of a lattice constructed in this manner:

Lemma 4.5.4. Let L'_{0,z_0} be a sublattice of L_{0,z_0} and define $L' = L'_{0,z_0} \cap L_{1,z_1} \cap \ldots \cap L_{n,z_n}$. Then $\dim L' = \dim L'_{0,z_0}$ and

$$\det L'_S \leq \det(L'_{0,z_0})_S \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}.$$

Proof. Fix a value of i, $1 \leq i \leq n$, and let $h \in \mathbb{Z}[x]$ correspond to a vector in L'_{0,z_0}. Then the definition of the lattice L_{i,z_i} and Lemma 4.4.1 imply that the vector corresponding to h belongs to $L'_{0,z_0} \cap L_{i,z_i}$ if and only if $h \in \mathfrak{p}_i^{z_i}$. Express h in the form

$$h = a_{z_i} \cdot (x - r_i)^{z_i} + \sum_{k=0}^{z_i - 1} a_k \cdot (x - r_i)^k,$$

where $a_0, \ldots, a_{z_i-1} \in \mathbb{Z}$ and $a_{z_i} \in \mathbb{Z}[x]$. Then $h \in \mathfrak{p}_i^{z_i}$ whenever $a_k \equiv 0 \pmod{p_i^{z_i-k}}$, for $0 \leq k \leq z_i - 1$. Thus,

$$\left[L'_{0,z_0} : L'_{0,z_0} \cap L_{i,z_i}\right] \leq \prod_{k=0}^{\min\{z_i,l\}} p_i^{z_i-k} \leq p_i^{\binom{z_i+1}{2}}.$$

Since i was arbitrary, it follows that

$$\left[L'_{0,z_0} : L'\right] = \left[L'_{0,z_0} : \bigcap_{i=1}^{n} \left(L'_{0,z_0} \cap L_{i,z_i}\right)\right] \leq \prod_{i=1}^{n} \left[L'_{0,z_0} : L'_{0,z_0} \cap L_{i,z_i}\right] \leq \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}.$$

Therefore, L' is a full-rank sublattice of L'_{0,z_0}. Hence,

$$\det L'_S = [(L'_{0,z_0})_S : L'_S] \cdot \det(L'_{0,z_0})_S = [L'_{0,z_0} : L'] \cdot \det(L'_{0,z_0})_S \leq \det(L'_{0,z_0})_S \cdot \prod_{i=1}^{n} p_i^{\binom{z_i+1}{2}}. \qquad \square$$

Remark 4.5.5. The method described by Guruswami et al. [69, Appendix B] for computing a basis of the intersection of two lattices applies to full-rank lattices only. As a result, whenever L'_{0,z_0} is not a full-rank sublattice of L_{0,z_0}, a basis for $L'_{0,z_0} \cap L_{1,z_1} \cap \ldots \cap L_{n,z_n}$ cannot be computed by their method alone. Instead, the method proposed by Cohen [36, Exercise 18 of Chapter 4] for computing the intersection of two lattices of arbitrary rank may be used to address the problem.

Given a sublattice L'_{0,z_0} of L_{0,z_0}, Lemma 4.5.4 implies that the lattice $L' = L'_{0,z_0} \cap L_{1,z_1} \cap \ldots \cap L_{n,z_n}$ will satisfy the inequality $\det(L'_S)^{1/\dim L'_S} < \det(L_S)^{1/(l+1)}$ whenever

$$\det(L'_{0,z_0})_S < \det(L_S)^{\frac{\dim L'_{0,z_0}}{l+1}} \cdot \prod_{i=1}^{n} p_i^{-\binom{z_i+1}{2}}.$$

Therefore, $\det(L'_{0,z_0})_S^{1/\dim L'_{0,z_0}}$ should be as small as possible. In the special case where $z_0 = 1$, the problem of constructing a sublattice $L'_{0,1} \subset L_{0,1}$ such that $\det(L'_{0,1})_S^{1/\dim L'_{0,1}}$ is small has already been addressed, with some success, by nonlinear polynomial selection algorithms based on Montgomery's method (see Chapter 3). Motivated by this success, the lattice construction used in nonlinear algorithms is generalised so that sublattices $L'_{0,z_0} \subset L_{0,z_0}$ with $\det(L'_{0,z_0})_S^{1/\dim L'_{0,z_0}}$ small can be constructed for $z_0 \geq 1$. To begin, the construction used in nonlinear algorithms is briefly reviewed within the context of this section.

Nonlinear algorithms construct sublattices of $L_{0,1}$ with small determinants from "small" geometric progressions modulo N. Recall that a geometric progression (GP) of length l and ratio r modulo N, denoted by a vector $[c_0, \ldots, c_{l-1}]$, is an integer sequence with the property that $c_i \equiv c_0 r^i \pmod{N}$, for $0 \leq i < l$. Central to the construction of lattices for nonlinear algorithms is the observation that

$$L_{0,1} = \left\{ (a_0, \ldots, a_l) \in \mathbb{Z}^{l+1} \mid \sum_{i=0}^{l} a_i c_i \equiv 0 \bmod N \right\}, \tag{4.42}$$

for any length $l+1$ geometric progression $[c_0, \ldots, c_l]$ with ratio m modulo N, nonzero terms and $\gcd(c_0, N) = 1$. Given such a GP, the orthogonal lattice of $[c_0, \ldots, c_l]\mathbb{Z}$ (see Section 3.2.1) forms a sublattices of $L_{0,1}$. More generally, sublattices of $L_{0,1}$ are constructed by considering the orthogonal lattice of a lattice generated by multiple linearly independent geometric progressions. In either case, Theorem 3.2.6 shows that the size of the determinant of the orthogonal lattice depends on the terms of the geometric progressions and not on N itself. Therefore, sublattices of $L_{0,1}$ with small determinant are obtained from geometric progressions with small terms.

To generalise the method of lattice construction used in nonlinear algorithms, a characterisation of L_{0,z_0} analogous to (4.42) is required for $z_0 \neq 1$. To this end, the Hasse derivative [73] (also called the hyperderivative) is now introduced. For each integer $k \geq 0$, the *k-th Hasse derivative* $D^{(k)} : \mathbb{R}[x] \to \mathbb{R}[x]$ is defined by

$$D^{(k)} = \frac{1}{k!} \frac{d^k}{dx^k}. \tag{4.43}$$

It follows immediately from the definition that a polynomial $h \in \mathbb{Z}[x]$ belongs to the ideal $\langle x - m, N \rangle$

if and only if $(D^{(0)}h)(m) \equiv 0 \pmod{N}$. More generally, the following lemma provides a necessary and sufficient condition for h to belong to the ideal $\langle x-m, N\rangle^z$, for some $z \geq 1$:

Lemma 4.5.6. Let $h \in \mathbb{Z}[x]$ and $z \geq 1$ be an integer. Then $h \in \langle N, x-m\rangle^z$ if and only if $(D^{(k)}h)(m) \equiv 0 \pmod{N^{z-k}}$, for $0 \leq k < z$.

Proof. Let $h = \sum_{i=0}^{l} a_i x^i \in \mathbb{Z}[x]$ and suppose that $(D^{(k)}h)(m) \equiv 0 \pmod{N^{z-k}}$, for $0 \leq k < z$. Then

$$\begin{aligned} h = \sum_{i=0}^{l} a_i \left((x-m)+m\right)^i &= \sum_{i=0}^{l} a_i \sum_{k=0}^{i} \binom{i}{k} m^{i-k}(x-m)^k \\ &= \sum_{k=0}^{l} \left(\sum_{i=k}^{l} a_i \binom{i}{k} m^{i-k} \right) (x-m)^k = \sum_{k=0}^{l} (D^{(k)}h)(m)(x-m)^k, \end{aligned}$$

where each term $(D^{(k)}h)(m)(x-m)^k \in \langle x-m, N\rangle^z$, for $0 \leq k \leq l$. Thus, $h \in \langle x-m, N\rangle^z$. Conversely, suppose that $h \in \langle x-m, N\rangle^z$. Then Lemma 4.4.1 implies that h can be written as an integer linear combination of the polynomials

$$b_i = N^{z-i}\,(x-m)^i, \text{ for } 0 \leq i \leq z; \quad \text{and} \quad b_i = x^{i-z}(x-m)^z, \text{ for } i > z.$$

Therefore, the linearity of $D^{(k)}$ implies that the converse will hold if $(D^{(k)}b_i)(m) \equiv 0 \pmod{N^{z-k}}$, for $0 \leq k < z$ and all $i \geq 0$. For $0 \leq i \leq z$, the definition (4.43) of $D^{(k)}$ implies that

$$(D^{(k)}b_i)(m) = \begin{cases} N^{z-k} & \text{if } i = k, \\ 0 & \text{if } i \neq k. \end{cases}$$

Similarly, for all $i > z$, it follows from the definition of $D^{(k)}$ that $(D^{(k)}b_i)(m) = 0$, for $0 \leq k < z$. Hence, the converse holds. □

The following corollary to Lemma 4.5.6 provides a generalisation of the characterisation (4.42):

Corollary 4.5.7. For $0 \leq k \leq \min\{z_0 - 1, l\}$, let $[c_{k,0}, \ldots, c_{k,l-k}]$ be a GP with nonzero terms, ratio m modulo N^{z_0-k} and $\gcd(c_{k,0}, N) = 1$. Then

$$L_{0,z_0} = \left\{ (a_0, \ldots, a_l) \in \mathbb{Z}^{l+1} \mid \sum_{i=k}^{l} a_i \binom{i}{k} c_{k,i-k} \equiv 0 \bmod N^{z_0-k}, \text{ for } 0 \leq k \leq \min\{z_0-1, l\} \right\}.$$

Proof. Let $(a_0, \ldots, a_l) \in \mathbb{Z}^{l+1}$ and $h = \sum_{i=0}^{l} a_i x^i$. Then the definition of the lattice L_{0,z_0} and Lemma 4.4.1 imply that $(a_0, \ldots, a_l) \in L_{0,z_0}$ if and only if $h \in \langle x-m, N\rangle^{z_0}$. Suppose that geometric progressions $[c_{k,0}, \ldots, c_{k,l-k}]$, for $0 \leq k \leq \min\{z_0 - 1, l\}$, satisfy the conditions of the corollary.

Then

$$(D^{(k)}h)(m) = \sum_{i=k}^{l} a_j \binom{i}{k} m^{i-k} \equiv c_{k,0}^{-1} \sum_{i=k}^{l} a_i \binom{i}{k} c_{k,0} m^{i-k} \equiv c_{k,0}^{-1} \sum_{i=k}^{l} a_i \binom{i}{k} c_{k,i-k} \pmod{N^{z_0-k}},$$

for $0 \leq k \leq \min\{z_0 - 1, l\}$. Moreover, $(D^{(k)}h)(m) = 0$, for all $k > l$. Therefore, Lemma 4.5.6 implies that $h \in \langle x - m, N \rangle^{z_0}$ if and only if

$$\sum_{i=k}^{l} a_i \binom{i}{k} c_{k,i-k} \equiv 0 \pmod{N^{z_0-k}}, \quad \text{for } 0 \leq k \leq \min\{z_0 - 1, l\}. \qquad \square$$

The characterisation of L_{0,z_0} provided by Corollary 4.5.7 permits the method of lattice construction used in nonlinear algorithms to be naturally generalised. The generalisation begins with the construction of geometric progressions $\boldsymbol{c}_k = [c_{k,0}, \ldots, c_{k,l-k}]$ with nonzero terms, ratio m modulo N^{z_0-k} and $\gcd(c_{k,0}, N) = 1$, for each value of k contained in some subset $I \subseteq \{0, \ldots, \min\{z_0 - 1, l\}\}$. Then a sublattice $L'_{0,z_0} \subset L_{0,z_0}$ is defined as follows:

$$L'_{0,z_0} = \{(a_0, \ldots, a_l) \in L_{0,z_0} \mid \langle (a_k, \ldots, a_l), \boldsymbol{c}_k \cdot \beta_{k,l} \rangle = 0, \text{ for all } k \in I\}, \tag{4.44}$$

where $\beta_{k,l}$ is the $(l - k + 1) \times (l - k + 1)$ diagonal matrix of binomial coefficients defined by

$$\beta_{k,l} = \operatorname{diag}\left(\binom{k}{k}, \binom{k+1}{k}, \ldots, \binom{l}{k}\right), \quad \text{for all } 0 \leq k \leq l.$$

More generally, for each $k \in I$, a set C_k of length $l - k + 1$ geometric progressions with ratio m modulo N^{z_0-k} may be constructed. Then, for each value of $k \in I$, the requirement in (4.44) that $\langle (a_k, \ldots, a_l), \boldsymbol{c}_k \cdot \beta_{k,l} \rangle = 0$ is replaced by the requirement that

$$\langle (a_k, \ldots, a_l), \boldsymbol{c} \cdot \beta_{k,l} \rangle = 0, \text{ for all } \boldsymbol{c} \in C_k. \tag{4.45}$$

Furthermore, for each value of $k \in I$, the requirement that $\boldsymbol{c}_k = [c_{k,0}, \ldots, c_{k,l-k}]$ has nonzero terms and $\gcd(c_{k,0}, N) = 1$ is replaced by the requirement that, for all $(a_k, \ldots, a_l) \in \mathbb{Z}^{l-k+1}$, (4.45) implies $\sum_{i=k}^{l} a_i \binom{i}{k} m^{i-k} \equiv 0 \pmod{N^{z_0-k}}$. If C_k satisfies this property and, in addition, there exists a geometric progression $[c_0, \ldots, c_{l-k}] \in C_k$ with $\gcd(c_0, N) = 1$, then C_k is referred to as *admissible*. A set C_k that contains at least one geometric progression $[c_0, \ldots, c_{l-k}]$ with nonzero terms and $\gcd(c_0, N) = 1$ is admissible.

For lattices constructed from admissible sets of geometric progressions in the manner just described, the following generalisation of Theorem 3.2.6 provides an upper bound on their determinant:

Theorem 4.5.8. *For $0 \leq k \leq \min\{z_0 - 1, l\}$, let C_k be either the empty set or an admissible set of* *length $l-k+1$ geometric progressions with ratio m modulo N^{z_0-k}. Define*

$$C = \bigcup_{k=0}^{\min\{z_0-1,l\}} \left\{ \left(0,\dots,0,\binom{k}{k}c_0,\binom{k+1}{k}c_1,\dots,\binom{l}{k}c_{l-k}\right) \in \mathbb{Z}^{l+1} \mid [c_0,\dots,c_{l-k}] \in C_k \right\}, \quad (4.46)$$

If the elements of C are linearly independent, then

$$L'_{0,z_0} = \{\boldsymbol{x} \in L_{0,z_0} \mid \langle \boldsymbol{x},\boldsymbol{c}\rangle = 0, \text{ for all } \boldsymbol{c} \in C\}$$

is an $(l+1-|C|)$-dimensional sublattice of L_{0,z_0} and

$$\det(L'_{0,z_0})_S \le \left(\prod_{\boldsymbol{c}\in C} \|\boldsymbol{c}\|_{2,s^{-1}}\right)\left(\prod_{k=0}^{\min\{z_0-1,l\}} N^{(z_0-k)(1-|C_k|)}\right),$$

for any $s>0$ and $S = s^{-l/2}\cdot \operatorname{diag}(1,s,\dots,s^l)$.

Proof. For $0\le k\le \min\{z_0-1,l\}$, let C_k be either the empty set or an admissible set of length $l-k+1$ geometric progressions with ratio m modulo N^{z_0-k}. Define $C\subset\mathbb{Z}^{l+1}$ according to (4.46) and assume its elements are linearly independent. Then the elements of C form a basis of a $|C|$-dimensional lattice $\Lambda\subseteq\mathbb{Z}^{l+1}$. Thus,

$$L'_{0,z_0} = \left\{\boldsymbol{x}\in\mathbb{Z}^{l+1} \mid \langle\boldsymbol{x},\boldsymbol{c}\rangle = 0, \text{ for all } \boldsymbol{c}\in C\right\} \cap L_{0,z_0} = \left(\mathbb{Z}^{l+1}\cap E_\Lambda^\perp\right)\cap L_{0,z_0} = \Lambda^\perp \cap L_{0,z_0}.$$

Let $\boldsymbol{a}_i = (a_{i,0},\dots,a_{i,l}) \in \Lambda^\perp$ and define $h_i = \sum_{j=0}^{l} a_{i,j}x^j \in \mathbb{Z}[x]$, for $i=1,2$. Then the definition of L_{0,z_0} and Lemma 4.4.1 imply that $\boldsymbol{a}_1-\boldsymbol{a}_2 \in L'_{0,z_0}$ if and only if $h_1-h_2\in\langle N,x-m\rangle^{z_0}$. Furthermore, Lemma 4.5.6 implies that $h_1-h_2\in\langle N,x-m\rangle^{z_0}$ if and only if

$$(D^{(k)}h_1)(m) \equiv (D^{(k)}h_2)(m) \pmod{N^{z_0-k}}, \quad \text{for } 0\le k\le\min\{z_0-1,l\}. \quad (4.47)$$

Given a geometric progression $\boldsymbol{c} = [c_0,\dots,c_{l-k}]\in C_k$, the definition of Λ implies that

$$\langle (a_{i,k},\dots,a_{i,l}),\boldsymbol{c}\cdot\beta_{k,l}\rangle = \left\langle \boldsymbol{a}_i, \left(0,\dots,0,\binom{k}{k}c_0,\binom{k+1}{k}c_1,\dots,\binom{l}{k}c_{l-k}\right)\right\rangle = 0, \quad \text{for } i=1,2.$$

Consequently, for all k such that C_k is admissible, it follows that

$$(D^{(k)}h_i)(m) = \sum_{j=k}^{l} a_{i,j}\binom{j}{k}m^{j-k} \equiv 0 \pmod{N^{z_0-k}} \quad \text{for } i=1,2.$$

Therefore, if the congruence in (4.47) is satisfied for all values of k such that $|C_k|=0$, then $\boldsymbol{a}_1-\boldsymbol{a}_2\in L'_{0,z_0}$. Hence,

$$\left[\Lambda^\perp : L'_{0,z_0}\right] \le \prod_{k:|C_k|=0} N^{z_0-k} = \prod_{k:|C_k|=0} N^{(z_0-k)(1-|C_k|)}. \quad (4.48)$$

Let s be a positive real and $S = s^{-l/2} \cdot \operatorname{diag}(1, s, \ldots, s^l)$. It follows from (4.48) that L'_{0,z_0} is a full-rank sublattice of $\Lambda^\perp$. Thus $\dim L'_{0,z_0} = \dim \Lambda^\perp = l + 1 - |C|$ and

$$\det(L'_{0,z_0})_S = \left[\Lambda_S^\perp : (L'_{0,z_0})_S\right] \cdot \det \Lambda_S^\perp = \left[\Lambda^\perp : L'_{0,z_0}\right] \cdot \det \Lambda_S^\perp \leq \det \Lambda_S^\perp \cdot \prod_{k:|C_k|=0} N^{(z_0-k)(1-|C_k|)}.$$

Therefore, to complete the proof, it suffices to show that

$$\det \Lambda_S^\perp \leq \left(\prod_{\boldsymbol{c} \in C} \|\boldsymbol{c}\|_{2,s^{-1}}\right)\left(\prod_{k:|C_k|\neq 0} N^{(z_0-k)(1-|C_k|)}\right). \tag{4.49}$$

Suppose that C_k is non-empty, for some $0 \leq k \leq \min\{z_0-1, l\}$. Then C_k is admissible and thus contains at least one GP, say $\boldsymbol{c}_{k,1} = [c_{1,0}, \ldots, c_{1,l-k}]$, with the property that $\gcd(c_{1,0}, N) = 1$. Therefore, if C_k contains a second GP, say $\boldsymbol{c}_{k,2} = [c_{2,0}, \ldots, c_{2,l-k}]$, then the vectors corresponding to $\boldsymbol{c}_{k,1}$ and $\boldsymbol{c}_{k,2}$ in the basis C for Λ,

$$\boldsymbol{c}'_{k,i} = \left(0, \ldots, 0, \binom{k}{k} c_{i,0}, \binom{k+1}{k} c_{i,1}, \ldots, \binom{l}{k} c_{i,l-k}\right) \in C, \quad \text{for } i = 1, 2,$$

satisfy $\boldsymbol{c}'_{k,2} - (c_{2,0}c_{1,0}^{-1})\boldsymbol{c}'_{k,1} \equiv \boldsymbol{0} \pmod{N^{z_0-k}}$. It follows that $\prod_{k:|C_k|\neq 0} N^{(z_0-k)(|C_k|-1)}$ divides each $|C| \times |C|$ minor of any basis matrix for Λ. Hence, Lemma 3.2.1 and Lemma 3.2.2 imply that

$$\det \Lambda_S^\perp \leq |\det S| \cdot \det \Lambda_{S^{-1}} \cdot \prod_{k:|C_k|\neq 0} N^{(z_0-k)(1-|C_k|)} = \det \Lambda_{S^{-1}} \cdot \prod_{k:|C_k|\neq 0} N^{(z_0-k)(1-|C_k|)}.$$

Finally, (4.49) is obtained by using Hadamard's inequality (see [153, Section 1.3]) and the basis C for Λ to bound $\det \Lambda_{S^{-1}}$. □

Let C be defined as in Theorem 4.5.8. Then a simple calculation shows that

$$\prod_{\boldsymbol{c} \in C} \|\boldsymbol{c}\|_{2,s^{-1}} = \prod_{k=0}^{\min\{z_0-1,l\}} \prod_{\boldsymbol{c} \in C_k} s^{-\frac{k}{2}} \cdot \|\boldsymbol{c} \cdot \beta_{k,l}\|_{2,s^{-1}}.$$

Therefore, a sublattice $L'_{0,z_0} \subseteq L_{0,z_0}$ constructed by means of Theorem 4.5.8 will have the property that $\det(L'_{0,z})_S$ is small whenever $\|\boldsymbol{c} \cdot \beta_{k,l}\|_{2,s^{-1}}$ is small, for all $\boldsymbol{c} \in C_k$ and $0 \leq k \leq \min\{z_0-1, l\}$. Moreover, combining Lemma 4.5.4 and Theorem 4.5.8 provides a sufficient condition for the corresponding lattice $L' = L'_{0,z_0} \cap L_{1,z_1} \cap \ldots \cap L_{n,z_n}$ to satisfy the inequality $\det(L'_S)^{1/\dim L'} < \det(L_S)^{1/(l+1)}$. Hence, it remains to determine if there exist geometric progressions that meet the required conditions and whether they can be found efficiently.

4.5.3 A Multivariate Generalisation

The approach to polynomial generation introduced in Section 4.1 required the construction of a nonzero polynomial $h \in \langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$. Given such a polynomial, Lemma 4.1.1 then implies that

$$N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \text{ divides } \operatorname{Res}(f,h), \text{ for all nonzero primitive } f \in \mathbb{Z}[x]. \tag{4.50}$$

Therefore, Lemma 2.1.3 implies that any non-constant irreducible polynomial $f \in \mathbb{Z}[x]$ with $f(m) \equiv 0 \pmod{N}$ and $\|f\|_{2,s}^{\deg h} \cdot \|h\|_{2,s}^{\deg f} < N^{z_0} \cdot \prod_{i=1}^{n} p_i^{z_i \sigma^*(f,\mathfrak{p}_i)}$ divides h over $\mathbb{Q}$. That is, all such f are found by the approach of Section 4.1. However, it is clear the requirement that $h \in \langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$ is stronger than needed: it is sufficient for h to satisfy the (possibly weaker) requirement that (4.50) holds. Moreover, (4.50) is only required to hold for a subset of all nonzero primitive polynomials in $\mathbb{Z}[x]$, i.e., those non-constant irreducible polynomials $f \in \mathbb{Z}[x]$ of bounded degree and size with $f(m) \equiv 0 \pmod{N}$. It is also clear that the efficacy of the approach of Section 4.1 is largely determined the existence of small $h \in \langle N, x-m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$. Therefore, it may be beneficial to relax the requirements on h in the hope that it will allow smaller polynomials to be found.

In this section, the approach of Section 4.1 is generalised with the aim of weakening the requirements on the polynomial h. This is in contrast to the previous section, where further restrictions were imposed on h. Here, the requirements on h are relaxed by imposing further conditions on the form of the number field sieve polynomials that are found. Explicitly, an irreducible polynomial $f(x, y_0, \ldots, y_k) \in \mathbb{Z}[x][y_0, \ldots, y_k]$ is chosen and each polynomial is required to be of the form $f = f(x, \boldsymbol{a})$, for some vector $\boldsymbol{a} \in \mathbb{Z}^{k+1}$. Then (4.50) may be replaced by the weaker requirement that $h \in \mathbb{Z}[x][y_0, \ldots, y_k]$ satisfies

$$N^{\sigma^*(f(x,\boldsymbol{a}),\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f(x,\boldsymbol{a}),\mathfrak{p}_i)z_i} \text{ divides } \operatorname{Res}(f(x,\boldsymbol{a}), h(x,\boldsymbol{a})), \tag{4.51}$$

for all $\boldsymbol{a} \in \mathbb{Z}^{k+1}$ such that $f(x, \boldsymbol{a})$ is nonzero and primitive. The weakened requirement is satisfied by any polynomial $h \in \mathbb{Z}[x]$ that satisfies (4.50).

The introduction of the additional variables $y_0, \ldots, y_k$ necessitates modifications to the approach of Section 4.1. Informally, the new approach is described as follows:

1. Select an irreducible polynomial $f \in \mathbb{Z}[x][y_0, \ldots, y_k]$ such that $\deg_x f = d$; and positive reals $Y_0, \ldots, Y_k$.

2. Choose pairwise comaximal ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i, r_i}$, with $p_i \nmid N$; and positive integer weights $z_0, \ldots, z_n$.

3. Find *sufficiently many* nonzero polynomials $h_1, \ldots, h_t \in \mathbb{Z}[x][y_0, \ldots, y_k]$ such that

   (4.52) h_j satisfies (4.51) for all $\boldsymbol{a} \in \mathbb{Z}^{k+1}$ such that $f(x, \boldsymbol{a})$ is non-constant and primitive;

   (4.53) $\deg h_j(x, \boldsymbol{a}) \leq l$, for all $\boldsymbol{a} \in \mathbb{Z}^{k+1}$;

(4.54) $\|h_j(x,a_0,\ldots,a_k)\|_{2,s}$ is small for all $(a_0,\ldots,a_k)\in\mathbb{Z}^{k+1}$ with $|a_i|\le Y_i$, for $1\le i\le k$.

4. Using the polynomials $f,h_1,\ldots,h_t$, find all irreducible polynomials of the form $f(x,\boldsymbol{a})$, for some $\boldsymbol{a}\in\mathbb{Z}^{k+1}$, such that

$$N^{\sigma^*(f(x,\boldsymbol{a}),\langle N,x-m\rangle)z_0}\cdot\prod_{i=1}^{n}p_i^{\sigma^*(f(x,\boldsymbol{a}),\mathfrak{p}_i)z_i} > \|f(x,\boldsymbol{a})\|_{2,s}^{l}\cdot\|h_j(x,\boldsymbol{a})\|_{2,s}^{d}, \quad \text{for } 1\le j\le t. \tag{4.55}$$

There are three main problems that arise from this approach: how to appropriately choose the polynomial $f\in\mathbb{Z}[x][y_0,\ldots,y_k]$ and the bounds $Y_0,\ldots,Y_k$; given f, how to construct sufficiently many polynomials $h_1,\ldots,h_t\in\mathbb{Z}[x][y_0,\ldots,y_k]$ satisfying properties (4.52)–(4.54); and finally, how to find all irreducible polynomials $f(x,\boldsymbol{a})$ that satisfy (4.55), given the polynomials $f,h_1,\ldots,h_t$. These problems are addressed (in reverse order) in the remainder of this section. The techniques used share much in common with algorithms based on Howgrave-Graham's reformulation [78] of Coppersmith's method [43] for finding small modular and integer roots of polynomial equations (see surveys of these methods and their applications by Bernstein [18] and May [117]). Concepts from Coppersmith's method have previously been applied to polynomial generation by Herrmann, May and Ritzenhofen [75]. To begin, the problem of finding all irreducible polynomials $f(x,\boldsymbol{a})$ that satisfy (4.55), is considered.

For nonzero polynomials $h_1,\ldots,h_t\in\mathbb{Z}[x][y_0,\ldots,y_k]$ that satisfy properties (4.52)–(4.54), and any $\boldsymbol{a}\in\mathbb{Z}^{k+1}$ such that $f(x,\boldsymbol{a})$ is irreducible and (4.55) holds, Lemma 4.1.2 implies that

$$\mathrm{Res}(f(x,\boldsymbol{a}),h_1(x,\boldsymbol{a}))=\mathrm{Res}(f(x,\boldsymbol{a}),h_2(x,\boldsymbol{a}))=\ldots=\mathrm{Res}(f(x,\boldsymbol{a}),h_t(x,\boldsymbol{a}))=0. \tag{4.56}$$

Therefore, by using resultants to successively eliminate the variables $y_0,\ldots,y_k$, a polynomial $R\in\mathbb{Z}[x]$ can be found for which all such $f(x,\boldsymbol{a})$ divide R over $\mathbb{Q}$. This method requires that there are at least $k+2$ algebraically independent polynomials among $f,h_1,\ldots,h_t$. For example, suppose $k=2$ and nonzero polynomials $h_1,\ldots,h_3\in\mathbb{Z}[x][y_0,y_1,y_2]$ have been found such that $f,h_1,\ldots,h_3$ are algebraically independent. Then resultant polynomials $R_1,\ldots,R_6\in\mathbb{Z}[x][y_0,y_1,y_2]$ can be computed:

$$\begin{array}{lclcl}
R_1(x,y_0,y_1)=\mathrm{Res}_{y_2}(f,h_1) & & & & \\
 & \Bigg\} & R_4(x,y_0)=\mathrm{Res}_{y_1}(R_1,R_2) & & \\
R_2(x,y_0,y_1)=\mathrm{Res}_{y_2}(f,h_2) & & & \Bigg\} & R_6(x)=\mathrm{Res}_{y_0}(R_4,R_5) \\
 & \Bigg\} & R_5(x,y_0)=\mathrm{Res}_{y_1}(R_2,R_3) & & \\
R_3(x,y_0,y_1)=\mathrm{Res}_{y_2}(f,h_3) & & & &
\end{array}$$

Algebraic independence implies that $R_6\neq 0$. Furthermore, for all $\boldsymbol{a}\in\mathbb{Z}^3$ such that $f(x,\boldsymbol{a})$ is irreducible and (4.56) holds, properties (4.6) and (4.10) imply that $f(x,\boldsymbol{a})$ divides $R_j(x,\boldsymbol{a})$ over $\mathbb{Q}$, for $1\le j\le 6$. Therefore, all such $f(x,\boldsymbol{a})$ can be found by factoring R_6 over $\mathbb{Q}$.

In the above example, the (total) degree of the resultant polynomials $R_1,\ldots,R_6$ may increase rapidly as each of the variables y_0, y_1 and y_2 are eliminated. Therefore, the use of resultants to address the third problem may require substantial time and memory. In practice, it may be beneficial to instead use Gröbner bases (see [46, Chapter 2]) to address this problem. This avenue will not be explored here. Instead, the reader is referred to Cox, Little and O'Shea [46, Chapter 3] and to a related application of Gröbner bases by Jochemsz and May [82, Section 6].

The polynomials $h_1, \ldots, h_t$ may be constructed by imitating ideas from Section 4.4 and Coppersmith's method, such as using lattice reduction reduction to find a small polynomial within a given ideal. Define ideals $\mathfrak{P}_0, \ldots, \mathfrak{P}_n \subseteq \mathbb{Z}[x][y_0, \ldots, y_k]$ as follows: $\mathfrak{P}_0 = \langle N, x - m\rangle$; and $\mathfrak{P}_i = \mathfrak{p}_i\mathbb{Z}[x][y_0, \ldots, y_k]$, for $1 \leq i \leq n$. Then, given a polynomial $h \in (\langle f\rangle + \mathfrak{P}_i)^{z_i}$, for some $1 \leq i \leq n$, it follows that $h(x, \boldsymbol{a}) \in \mathfrak{p}_i^{z_i\sigma(f(x,\boldsymbol{a}),\mathfrak{p}_i)}$, for all $\boldsymbol{a} \in \mathbb{Z}^{k+1}$. Therefore, Lemma 4.1.1 implies that (4.52) is satisfied by all $h_j \in \prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$. This choice of ideal directly generalises ideals from Coppersmith's method. However, by exploiting properties of resultants, a suitable choice of ideal that is larger (in the sense that it contains $\prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$) may be obtained:

Lemma 4.5.9. Let N be a nonzero integer; $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subset \mathbb{Z}[x]$ be pairwise comaximal ideals of the form $\mathfrak{p}_i = \mathfrak{p}_{p_i,r_i}$, with $p_i \nmid N$, for $1 \leq i \leq n$; and $z_0, \ldots, z_n$ be positive integers such that $z_i = z_j$ whenever $p_i = p_j$, for $1 \leq i < j \leq n$. Suppose that $f \in \mathbb{Z}[x]$ is non-constant, primitive and $\gcd(\mathrm{lc}(f), N) = 1$. Then, for all $h \in \langle f\rangle + \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$,

$$N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i:p_i\nmid \mathrm{lc}(f)} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \cdot \prod_{\substack{i:p_i\mid \mathrm{lc}(f)\\ f'\notin\mathfrak{p}_i}} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \text{ divides } \mathrm{Res}(f, h) \text{ in } \mathbb{Z}. \tag{4.57}$$

Proof. Suppose that $f \in \mathbb{Z}[x]$ is non-constant, primitive and $\gcd(\mathrm{lc}(f), N) = 1$. Then, given a polynomial $h \in \langle f\rangle + \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$, there exist polynomials $a, b \in \mathbb{Z}[x]$ such that $h = a \cdot f + b$ and $b \in \langle N, x - m\rangle^{z_0} \cdot \prod_{i=1}^{n} \mathfrak{p}_i^{z_i}$. Therefore, (2.3) implies that

$$\mathrm{Res}(f, h) = \mathrm{lc}(f)^{\deg h} \cdot \prod_{\alpha:f(\alpha)=0} h(\alpha) = \mathrm{lc}(f)^{\deg h} \cdot \prod_{\alpha:f(\alpha)=0} b(\alpha) = \mathrm{lc}(f)^{\deg h-\deg b} \cdot \mathrm{Res}(f, b). \tag{4.58}$$

Moreover, since f is non-constant and primitive, Lemma 4.1.1 implies that

$$N^{\sigma^*(f,\langle N,x-m\rangle)z_0} \cdot \prod_{i=1}^{n} p_i^{\sigma^*(f,\mathfrak{p}_i)z_i} \text{ divides } \mathrm{Res}(f, b) \text{ in } \mathbb{Z}. \tag{4.59}$$

Suppose there exist $t \geq 1$ distinct indices $1 \leq i_1, \ldots, i_t \leq n$ such that $f \in \mathfrak{p}_{i_j}$ and $f' \notin \mathfrak{p}_{i_j}$, for $1 \leq j \leq t$; and $p_{i_1} = \ldots = p_{i_t}$. Then $\sigma^*(f, \mathfrak{p}_{i_j}) = 1$, for $1 \leq j \leq t$ (cf. the proof of Lemma 4.4.6). Moreover, Hensel's lemma (see [36, Theorem 3.5.3]) implies that there exist integers $\bar{r}_{i_1}, \ldots, \bar{r}_{i_t}$ such that $f(\bar{r}_{i_j}) \equiv 0 \pmod{p_{i_j}^{z_{i_j}}}$ and $\bar{r}_{i_j} \equiv r_{i_j} \pmod{p_{i_j}}$, for $1 \leq j \leq t$. Therefore, the assumption that the ideals $\mathfrak{p}_{i_1}, \ldots, \mathfrak{p}_{i_t}$ are pairwise comaximal implies that

$$\gcd\left(p_{i_j}^{z_{i_j}}, \bar{r}_{i_j} - \bar{r}_{i_k}\right) = 1, \quad \text{for all } 1 \leq j < k \leq t.$$

In particular, this implies that the ideals $\langle p_{i_j}^{z_{i_j}}, x - \bar{r}_{i_j}\rangle$, for $1 \leq j \leq t$, are pairwise comaximal. Finally, since $b \in \bigcap_{i=1}^{n} \mathfrak{p}_i^{z_i}$, the congruence $\bar{r}_{i_j} \equiv r_{i_j} \pmod{p_{i_j}}$ implies that $b \in \langle p_{i_j}^{z_{i_j}}, x - \bar{r}_{i_j}\rangle$, for $1 \leq j \leq t$. As a result, $f, h \in \prod_{j=1}^{t}\langle p_{i_j}^{z_{i_j}}, x - \bar{r}_{i_j}\rangle$, where $p_{i_1}^{z_{i_1}} = \ldots = p_{i_t}^{z_{i_t}}$. Therefore, using the (established) claim made at the beginning of the proof of Lemma 4.1.1 (see Section 4.2.2), it follows that $\prod_{j=1}^{t} p_{i_j}^{z_{i_j}}$ divides $\mathrm{Res}(f,h)$ in $\mathbb{Z}$. Hence,

$$\prod_{i: f' \notin \mathfrak{p}_i} p_i^{\sigma^*(f,\mathfrak{p}_i) z_i} \text{ divides } \mathrm{Res}(f,h) \text{ in } \mathbb{Z}. \tag{4.60}$$

Consequently, (4.57) is obtained by combining (4.58), (4.59) and (4.60) with the assumption that $\gcd(\mathrm{lc}(f), N) = 1$. □

Suppose that $z_0, \ldots, z_n$ satisfy $z_i = z_j$ whenever $p_i = p_j$, for $1 \leq i < j \leq n$. Then Lemma 4.5.9 implies that (4.52) is *almost* satisfied by all $h_j \in \langle f\rangle + \prod_{i=0}^{n} \mathfrak{P}_i^{z_i}$: there is the additional requirement that $\gcd(\mathrm{lc}(f(x,\boldsymbol{a})), N) = 1$ and the contributions of some roots modulo primes that divide $\mathrm{lc}(f(x,\boldsymbol{a}))$ are potentially lost. In practice, the requirement that $\gcd(\mathrm{lc}(f(x,\boldsymbol{a})), N) = 1$ is nonrestrictive. Moreover, the additional requirement that the contribution of distinct roots modulo the same prime must be weighted equally appears to be natural in the setting of polynomial generation (cf. Section 4.4.1). Therefore, the expense, if any, for which the ideal $\langle f\rangle + \prod_{i=0}^{n} \mathfrak{P}_i^{z_i}$ can be used may be outweighed by the fact that it is larger than $\prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$. This avenue requires further investigation. In particular, it is unknown if Lemma 4.5.9 may be improved upon. Therefore, for the remainder of the section, it is assumed that polynomials $h_1, \ldots, h_t \in \prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$ are constructed.

Let $\mathcal{M}$ be a set of monomials in the variables $x, y_0, \ldots, y_k$. Then a polynomial $g \in \mathbb{Z}[x][y_0, \ldots, y_k]$ is said to be *defined over* $\mathcal{M}$ if and only if g can expressed as an integer linear combination of monomials in $\mathcal{M}$. Suppose that $\mathcal{M}$ is finite and $L \subseteq \mathbb{Z}^{|\mathcal{M}|}$ contains a coefficient vector for each polynomial $h \in \prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$ such that h is defined over $\mathcal{M}$. Then L forms an $|\mathcal{M}|$-dimensional lattice. The following lemma shows that polynomials $h_1, \ldots, h_t \in \prod_{i=0}^{n}(\langle f\rangle + \mathfrak{P}_i)^{z_i}$ that satisfy (4.53) and (4.54) may be found by searching for short vectors in an appropriate rescaling of L:

Lemma 4.5.10. Let $h \in \mathbb{Z}[x][y_0, \ldots, y_k]$ have degree l in x. Suppose that $h = \sum_{i=0}^{l} h_i x^i$, where each coefficient $h_i \in \mathbb{Z}[y_0, \ldots, y_k]$ contains at most ω distinct monomials, for $0 \leq i \leq l$. Then

$$\|h(x, a_0, \ldots, a_k)\|_{2,s} \leq \sqrt{\omega} \cdot s^{-\frac{l}{2}} \cdot \|h(xs, y_0 Y_0, \ldots, y_k Y_k)\|_2\,,$$

for all $(a_0, \ldots, a_k) \in \mathbb{Z}^{k+1}$ such that $|a_i| \leq Y_i$, for $0 \leq i \leq k$, and $s > 0$.

As part of the proof of Lemma 4.5.10, arguments of Howgrave-Graham [78] are used to bound the coefficients of $h(x, a_0, \ldots, a_k)$, for all $(a_0, \ldots, a_k) \in \mathbb{Z}^{k+1}$ such that $|a_i| \leq Y_i$, for $0 \leq i \leq k$. This part of the proof is included here for the sake of completeness.

Proof. For $0 \leq i \leq l$, suppose that $h_i = \sum_{j_0, \ldots, j_k} h_{i,j_0,\ldots,j_k} y_0^{j_0} \cdots y_k^{j_k}$, where the coefficients $h_{i,j_0,\ldots,j_k} \in$

$\mathbb{Z}$. Furthermore, let $(a_0, \ldots, a_k) \in \mathbb{Z}^{k+1}$ satisfy $|a_i| \leq Y_i$, for $0 \leq i \leq k$. Then

$$
\begin{aligned}
|h_i(a_0, \ldots, a_k)| &\leq \sum_{j_0, \cdots, j_k} \left| h_{i,j_0,\ldots,j_k} a_0^{j_0} \cdots a_k^{j_k} \right| = \sum_{j_0, \cdots, j_k} \left| h_{i,j_0,\ldots,j_k} Y^{j_0} \cdots Y^{j_k} \left(\frac{a_0}{Y_0}\right)^{j_0} \cdots \left(\frac{a_k}{Y_k}\right)^{j_k} \right| \\
&\leq \sum_{j_0, \cdots, j_k} |h_{i,j_0,\ldots,j_k}| \, Y^{j_0} \cdots Y^{j_k} \leq \sqrt{\omega} \cdot \|h_i(y_0 Y_0, \ldots, y_k Y_k)\|_2 \,,
\end{aligned}
$$

for $0 \leq i \leq l$. Hence,

$$
\begin{aligned}
\|h(x, a_0, \ldots, a_k)\|_{2,s} &= \sqrt{\sum_{i=0}^{l} |h_i(a_0, \ldots, a_k)|^2 s^{2i-l}} \leq \sqrt{\sum_{i=0}^{l} \omega \, \|h_i(y_0 Y_0, \ldots, y_k Y_k)\|_2^2 \, s^{2i-l}} \\
&= \sqrt{\omega} \cdot s^{-\frac{l}{2}} \cdot \|h(xs, y_0 Y_0, \ldots, y_k Y_k)\|_2 \,,
\end{aligned}
$$

for all $s > 0$. □

Lemma 4.5.10 suggests the following strategy for constructing polynomials $h_1, \ldots, h_t \in \mathbb{Z}[x][y_0, \ldots, y_k]$ that satisfy properties (4.52)–(4.54):

1. Select a finite set of monomials $\mathcal{M}$ in the variables $x, y_0, \ldots, y_k$ such that $t \leq |\mathcal{M}|$ and $\deg_x \mu \leq l$, for all $\mu \in \mathcal{M}$.
2. Compute a basis for a lattice $L \subseteq \mathbb{Z}^{|\mathcal{M}|}$ that consists entirely of coefficient vectors from those polynomials in $\prod_{i=0}^{n} (\langle f \rangle + \mathfrak{P}_i)^{z_i}$ that are defined over $\mathcal{M}$.
3. Let L_S be the lattice obtained by rescaling the coordinate of L corresponding to the monomial $x^j y_0^{j_0} \cdots y_k^{j_k} \in \mathcal{M}$ by $s^{j-l/2} Y_0^{j_0} \cdots Y_k^{j_k}$. Use lattice reduction to find a basis for L_S consisting of short vectors and return the corresponding polynomials $h_1, \ldots, h_t$ corresponding to the t shortest basis vectors.

Polynomials $h_1, \ldots, h_t$ constructed in this manner are guaranteed to be linearly independent. However, the polynomials mail fail to be algebraically independent, possibly preventing the elimination of variables by resultant computations. The same obstruction arises in multivariate generalisations of Coppersmith's method. In that setting, some applications of the method have continued to work in practice without modification. However, it has been demonstrated by Blömer and May [21] and Hinek [76, 77] that this is not always the case. It may be that good fortune is found in this setting and algebraic independence occurs frequently. However, this possibility must be verified in practice by experiments. Progress toward a rigorous multivariate generalisation of Coppersmith's method has been made by Bauer and Joux [13]. Ideas from their approach may transfer over to this setting. The problem of generating algebraically independent polynomials is not considered further here.

The selection of the polynomial $f \in \mathbb{Z}[x][y_0, \ldots, y_k]$ and the corresponding bounds $Y_0, \ldots, Y_k$ is now briefly considered. A straight forward choice is to take $f = \sum_{i=0}^{d} y_i x^i$. Then a bound M on $\|f(x, \boldsymbol{a})\|_{2,s}$

can be chosen and the bounds $Y_0, \ldots, Y_d$ defined by $Y_i = Ms^{d/2-i}$, for $0 \leq i \leq d$. However, for lattices with small determinant to be obtained, the bounds $Y_0, \ldots, Y_k$ should be as small as possible. Thus, a better approach is to exploit the fact that polynomials with root m modulo N are sought, by setting

$$f(x, y_0, \ldots, y_d) = \sum_{i=1}^{d} y_i(x^i - m^i) + y_0 N.$$

Then a polynomial $g \in \mathbb{Z}[x]$ of degree at most d satisfies $g(m) \equiv 0 \pmod{N}$ if and only if $g = f(x, \boldsymbol{a})$, for some $\boldsymbol{a} \in \mathbb{Z}^{d+1}$. Moreover, $f(m, a_0, \ldots, a_d) = a_0 N$, for all $(a_0, \ldots, a_d) \in \mathbb{Z}^{d+1}$. Therefore, the bound Y_0 can be significantly reduced, with the remaining bounds $Y_0, \ldots, Y_{d-1}$ remaining unchanged. To help obtain polynomials with good root properties, f may be modified by selecting two products of small prime powers $a, b \in \mathbb{Z}$ and setting

$$f(x, y_0, \ldots, y_d) = a y_d(x^d - m^d) + \sum_{i=1}^{d-1} y_i(x^i - m^i) + y_0 bN.$$

Accordingly, the bounds Y_0 and Y_d may be reduced by factors of $|a|$ and $|b|$ respectively. A final modification is to assign values to y_0 or y_d, thus reducing the number of variables.

Determining the viability of the approach introduced in this section requires filling in many of the details missing from the outline provided here. In particular, those details relating to the construction of the polynomials $h_1, \ldots, h_t$. The approach generalises concepts from Howgrave-Graham's reformulation of Coppersmith's method. As a result, there is potential for the methods discussed in this section to be utilised in new attacks on RSA, and in addressing factorisation problems (see [18, 117]). Such an application of the methods may in practice be conceptually simpler. Consequently, furthering their development in such a setting before attempting to apply them to polynomial generation may be worthwhile.

Chapter 5

Smooth Elements in Number Fields

In Chapter 4, ideas extracted from the framework for list decoding of algebraic error-correcting codes were used to develop a new approach to polynomial selection. Motivation for the approach was provided in part by previous applications of list decoding algorithms in number theory. In particular, the following two examples were cited: Cheng and Wan's [35] demonstration that a list decoding algorithm for Reed–Solomon codes can be used to find smooth polynomials over finite fields, and Boneh's [25] use of a list decoding algorithm for Chinese remainder codes to find smooth integers. Both these examples provide strong evidence toward the utility of applying list decoding algorithms for algebraic error-correcting codes to problems of finding elements in a ring with a smooth representation. Further evidence toward this claim is provided in this chapter. Here a list decoding algorithm is developed and used to generalise Boneh's result to algebraic number fields.

Error-correcting codes derived from algebraic number fields were first considered by Lenstra [106] and more recently by Guruswami [67]. These codes generalise the construction of Chinese remainder codes (or simply, CRT codes), the number-theoretic analogues of Reed-Solomon codes. Although the two are similar, Guruswami's construction of number field codes, called *NF-codes*, is less general than the construction given by Lenstra. However, the generalisation of CRT codes to NF-codes provides a clearer analogue of the generalisation to algebraic-geometry codes of Reed-Solomon codes.

Currently, all known algorithms for decoding of codes derived from algebraic number fields are limited to CRT codes. For CRT codes, decoding reduces to the following problem: given n relatively prime integers $p_1 < \ldots < p_n$, a vector $(r_1, \ldots, r_n) \in \mathbb{Z}^n$, and an integer $k < n$, find all $m \in \mathbb{Z}$ with $0 \leq m < \prod_{i=1}^{k} p_i$ such that $m \equiv r_i \pmod{p_i}$ for t values of i, $1 \leq i \leq n$. For $t \geq (n+k)/2$, if such a value of m exists, then it is unique and can be found with Mandelbaum's [113] decoding algorithm. For smaller values of t, uniqueness is no longer guaranteed and the problem is referred to as the *list decoding* problem for CRT codes. The first efficient algorithm for list decoding of CRT codes was provided by Goldreich, Ron and Sudan [62]. Their algorithm runs in polynomial time and solves the problem whenever

$$t \geq \left(1 + \frac{2}{k}\right) \cdot \sqrt{2kn\frac{\log p_n}{\log p_1}} + \frac{k+6}{2}.$$

Subsequently, Boneh [25] improved upon the algorithm of Goldreich, Ron and Sudan by providing a polynomial time algorithm which solves the problem whenever $t \geq \sqrt{kn \log p_n / \log p_1}$.

A common problem with these algorithms is a decline in decoding ability whenever $p_1 \ll p_n$. In the case of Mandelbaum's algorithm, this may cause the algorithm to no longer run in polynomial time (see [62]). Roughly speaking, this problem occurs since the residues of an integer m modulo small p_i provide less information than the residues of m modulo large p_i. This problem was overcome for unique and list decoding by Guruswami, Sahai and Sudan [69] by weighting the contribution of each p_i. This led to the first polynomial time algorithm capable of decoding for all $t \geq (n+k)/2$. In order to overcome the problem for list decoding, Guruswami et al. gave an efficient algorithm for solving the more general problem of weighted list decoding of CRT codes. Given positive weights w_i assigned to the p_i, the *weighted list decoding* problem for CRT codes asks to find all $m \in \mathbb{Z}$ with $0 \leq m < \prod_{i=1}^{k} p_i$ such that $\sum_i w_i \geq t$, where the sum is over all i such that $m \equiv r_i \pmod{p_i}$. The list decoding problem for CRT codes then corresponds to the case where $w_i = 1$, for $1 \leq i \leq n$. By carefully selecting weights, the algorithm of Guruswami et al. can solve the list decoding problem for CRT codes whenever $t \geq \sqrt{k(n+\varepsilon)}$, for arbitrarily small $\varepsilon > 0$, in time polynomial in n, $\log p_n$ and $1/\varepsilon$. As a result, the decoding performance of the algorithm essentially matches that of the celebrated list decoding algorithms for Reed-Solomon and algebraic geometry codes [159, 70].

It is natural to ask whether decoding algorithms exist for codes constructed from arbitrary number fields. The construction of NF-codes lies within the general framework of "ideal-based" codes [69, 160], thus lending itself to decoding by an algorithm based on the framework for list decoding of algebraic error-correcting described by Guruswami et al. [69, Appendix A]. In this chapter, this observation is used to generalise the algorithm of Guruswami et al. for list decoding of CRT codes to number fields, resulting in the first algorithm for solving the weighted list decoding problem for NF-codes. The decoding algorithm then plays a central role in the development of an algorithm for finding algebraic integers in a number field with norm containing a large smooth factor. Finally, two different approaches are used to derive new bounds on the existence of such elements. The first approach, uses combinatorial arguments based on generic coding bounds. The second, generalises Boneh's algorithmically derived bounds on the number of smooth integers in short intervals.

The remainder of the chapter is organised as follows. In Section 5.1, relevant background material on NF-codes is provided and notation established. In Section 5.2, generic coding bounds are used to obtain conditions under which decoding of NF-codes is combinatorially feasible. In Section 5.3 and Section 5.3.2, an algorithm for weighted list decoding of NF-codes is developed and analysed. Similarly, Section 5.3.3 is dedicated to the development and analysis of a computationally simpler version of the algorithm. Section 5.3.4 focuses on choosing parameters for the decoding algorithm and the comparison of its performance against the theoretical bounds of Section 5.2. The coding theory portion of this chapter is completed in Appendix A.1 with the derivation of a new family of codes constructed from number fields for which their rate is easily computable. Finally, Section 5.4 contains results on smooth elements in number fields.

While this chapter was in preparation, the work of Cohn and Heninger [38] came to the author's attention. There an algorithm for solving polynomial equations over number fields is provided. Their algorithm shares much in common with algorithms presented in this chapter and leads to an alternative approach to list decoding of NF-codes (see Remark 5.3.7). In this application, both algorithms yield similar results.

5.1 Review of NF-codes

Introduced by Guruswami [67], NF-codes serve as a natural generalisation of CRT codes to number fields. In this section, their construction is briefly reviewed and the weighted list decoding problem for NF-codes introduced. For further background and motivation behind the construction of NF-codes, the reader is referred to Guruswami's original description. To begin, notation that is used throughout the remainder of the chapter is introduced.

Throughout the chapter, K denotes a number field of degree d and signature (r_1, r_2). Denote by $\mathcal{O}_K$ the ring of integers in K; by D_K its discriminant; and by $\omega_1, \ldots, \omega_d$ an integral basis for $\mathcal{O}_K$. Let $\sigma_1, \ldots, \sigma_d$ denote the field embeddings of K in the field $\mathbb{C}$, ordered such that $\sigma_1, \ldots, \sigma_{r_1}$ are the real embeddings of K, and the complex embeddings $\sigma_{r_1+1}, \ldots, \sigma_d$ of K satisfy $\sigma_{r_1+i} = \overline{\sigma_{r_1+r_2+i}}$, for $1 \leq i \leq r_2$. For all $x \in K$, the *(field) norm* of x, denoted $N_K(x)$, is defined as the product $N_K(x) = \prod_{i=1}^{d} \sigma_i(x)$. For any nonzero ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, the quotient $\mathcal{O}_K/\mathfrak{a}$ is finite. The *norm* of a nonzero integral ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, denoted $\mathfrak{N}\mathfrak{a}$, is defined by $\mathfrak{N}\mathfrak{a} = |\mathcal{O}_K/\mathfrak{a}|$. For all $x \in \mathcal{O}_K$, the relationship $|N_K(x)| = \mathfrak{N}\langle x \rangle$ holds, where $\langle x \rangle$ denotes the principal ideal generated by x in $\mathcal{O}_K$. Given a nonzero integral ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, it follows that $\mathfrak{N}\mathfrak{a}$ divides $N_K(x)$, for all $x \in \mathfrak{a}$. For background on algebraic number theory, the reader is referred to the texts of Marcus [114] and Narkiewicz [128].

Before defining NF-codes, the following notion of the size of an element in K requires introduction: given a vector $\boldsymbol{s} = (s_1, \ldots, s_{r_1+r_2}) \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$, the *$\boldsymbol{s}$-shifted size* of an element $x \in K$ is defined as

$$\mathrm{size}_{\boldsymbol{s}}(x) = \sum_{i=1}^{r_1} |\sigma_i(x) - s_i| + 2 \sum_{i=1}^{r_2} |\sigma_{r_1+i}(x) - s_{r_1+i}|.$$

The $\mathbf{0}$-shifted size of an element $x \in K$ is simply denoted by $\mathrm{size}(x)$. For $x, y \in K$ and $\boldsymbol{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$, it is readily verified that $\mathrm{size}(x - y) \leq \mathrm{size}_{\boldsymbol{s}}(x) + \mathrm{size}_{\boldsymbol{s}}(y)$. Moreover, an application of the AM-GM inequality shows that $|N_K(x)| \leq \mathrm{size}(x)^d / d^d$, for all $x \in K$.

Definition 5.1.1 (NF-codes)**.** Let K be a number field that contains pairwise relatively prime ideals

$\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subseteq \mathcal{O}_K$, ordered so that $\mathfrak{N}\mathfrak{p}_1 \leq \mathfrak{N}\mathfrak{p}_2 \leq \ldots \leq \mathfrak{N}\mathfrak{p}_n$. An *NF-code* $\mathcal{C} = \mathcal{C}_K$, based on K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$, is defined to be the set

$$\mathcal{C} = \left\{ (m + \mathfrak{p}_1, \ldots, m + \mathfrak{p}_n) \in \frac{\mathcal{O}_K}{\mathfrak{p}_1} \times \ldots \times \frac{\mathcal{O}_K}{\mathfrak{p}_n} \mid m \in \mathcal{O}_K \text{ and } \operatorname{size}_{\boldsymbol{s}}(m) \leq M \right\}.$$

The set $\mathcal{M}_{\mathcal{C}} = \{m \in \mathcal{O}_K \mid \operatorname{size}_{\boldsymbol{s}}(m) \leq M\}$ is referred to as the *message set* of $\mathcal{C}$.

The *information rate* (or simply, *rate*) of an NF-code $\mathcal{C}$ with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$ is defined to be the quotient $R(\mathcal{C}) = \log|\mathcal{M}_{\mathcal{C}}| / \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i$ (see [155, Section II]). The rate provides a measure of the amount of redundancy added by encoding a message $m \in \mathcal{M}_{\mathcal{C}}$ as the vector $(m+\mathfrak{p}_1, \ldots, m+\mathfrak{p}_n) \in \mathcal{C}$. Determining the cardinality of $\mathcal{M}_{\mathcal{C}}$, and thus the rate of $\mathcal{C}$, is a nontrivial problem. Guruswami [67, Section E] noted that a standard argument from the geometry of numbers suggests that $|\mathcal{M}_{\mathcal{C}}| \approx \frac{2^{r_1}\pi^{r_2}}{\sqrt{|D_K|}} \frac{M^d}{d!}$. However, this estimate cannot be used in general since the error term may dominate. Instead, Guruswami [67, Proposition 19] used an averaging argument due to Lenstra [106] to prove the existence of a shift $\boldsymbol{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ such that

$$\Big| \{x \in \mathcal{O}_K \mid \operatorname{size}_{\boldsymbol{s}}(x) \leq M\} \Big| \geq \frac{2^{r_1}\pi^{r_2}}{\sqrt{|D_K|}} \frac{M^d}{d!}. \tag{5.1}$$

For any such shift $\boldsymbol{s}$, an NF-code $\mathcal{C}$ based on K, with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M; \boldsymbol{s})$, has rate $R(\mathcal{C})$ satisfying

$$R(\mathcal{C}) \geq \frac{\log(2^{r_1}\pi^{r_2}M^d) - \log d! - \log\sqrt{|D_K|}}{n \log \mathfrak{N}\mathfrak{p}_n}.$$

The existence proof is nonconstructive and it remains an open problem as to how to find a vector $\boldsymbol{s}$ satisfying (5.1). As an aside, in Appendix A.1, the construction of NF-codes is modified to obtain a new family of codes, each with rate that is easily computable.

Given two vectors in an NF-code $\mathcal{C}$, their *Hamming distance* is the number of coordinates at which they differ. The *minimum distance* (or simply, *distance*) of $\mathcal{C}$ is then the minimum of the Hamming distance over all pairs of distinct vectors in $\mathcal{C}$. Throughout the chapter, given an ideal $\mathfrak{a} \subseteq \mathcal{O}_K$ and an element $x \in \mathcal{O}_K$, define $\sigma(x, \mathfrak{a}) = 1$, if $x \in \mathfrak{a}$; and $\sigma(x, \mathfrak{a}) = 0$, otherwise. Then the distance of an NF-code $\mathcal{C}$, with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$, is the minimum of the sum $\sum_{i=1}^{n}(1 - \sigma(x-y, \mathfrak{p}_i))$ over all pairs of distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$. To obtain a lower bound on distance of $\mathcal{C}$, denoted herein by $d(\mathcal{C})$, consider distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$. Then

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma(x-y,\mathfrak{p}_i)} \leq |N_K(x-y)| \leq \frac{1}{d^d}\operatorname{size}(x-y)^d \leq \frac{1}{d^d}\left(\operatorname{size}_{\boldsymbol{s}}(x) + \operatorname{size}_{\boldsymbol{s}}(y)\right)^d \leq \left(\frac{2M}{d}\right)^d. \tag{5.2}$$

Therefore, if there exists a value of $k \leq n$ such that $(2M/d)^d \leq \prod_{i=1}^{k} \mathfrak{N}\mathfrak{p}_i$, then $\sum_{i=1}^{n} \sigma(x-y, \mathfrak{p}_i) \leq k$. Since x and y were arbitrary distinct elements of $\mathcal{M}_{\mathcal{C}}$, it follows that $d(\mathcal{C}) \geq n - k$, for any value of $k \leq n$ such that $(2M/d)^d \leq \prod_{i=1}^{k} \mathfrak{N}\mathfrak{p}_i$.

For an NF-code $\mathcal{C}$, with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$, decoding reduces to the following problem: given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{p}_n$ and $t \geq 0$, find all $m \in \mathcal{M}_{\mathcal{C}}$ such that $\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) \geq t$. For $t \geq n - (d(\mathcal{C}) - 1)/2$, if such an element $m \in \mathcal{M}_{\mathcal{C}}$ exists, then it is unique. Accordingly, decoding for such values of t is referred to as *unique decoding.* For smaller values of t, uniqueness is no longer guaranteed and decoding referred to as *list decoding.* The notion of list decoding was introduced by Elias [53] and Wozencraft [169]. A discussion of the history of list decoding and its utility as a relaxation of unique decoding is provided by Guruswami [68, Section 1.3] (see also references therein). The final decoding paradigm considered in this chapter, called *weighted list decoding*, is a relaxation of traditional list decoding. In contrast to list decoding, weighted list decoding may not treat the contribution of each coordinate equally. Instead, the contributions of the coordinates are determined by individual weights. For an NF-code $\mathcal{C}$, with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$, weighted list decoding reduces to the following problem: given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{p}_n$, positive real weights $\beta_1, \ldots, \beta_n$, and $t \geq 0$, find all $m \in \mathcal{M}_{\mathcal{C}}$ such that $\sum_{i=1}^{n} \sigma(m-r_i, \mathfrak{p}_i)\beta_i \geq t$. List decoding is then captured by the special case in which all the weights are equal.

5.2 Combinatorial Bounds on List Decoding

An algorithm that performs weighted list decoding is required to return *all* codewords with sufficiently large weighted agreement. Therefore, a necessary condition for a decoding algorithm to run in polynomial time is that only polynomially many codewords are returned. The classical Johnson bound [83, 84] provides an upper bound on the number of codewords in a binary code at Hamming distance *exactly* e from an arbitrary word. This bound was later generalised by Guruswami and Sudan [71] who gave a "Johnson-type" bound for the number of codewords at distance *at most* e from an arbitrary word in a q-ary code. In addition, Guruswami and Sudan extended their result to provide bounds on the number of codewords with sufficiently large weighted agreement. In this section, a related result due to Guruswami [68, Theorem 7.10], namely Lemma 4.3.1, is used to derive combinatorial bounds on the decoding of NF-codes. As a result, the bounds obtained in this section are analogous to those obtain for number field sieve polynomial generation in Section 4.3. To begin, Lemma 4.3.1 is used to provide a bound on the number of codewords in an NF-code with sufficiently large weighted agreement.

Theorem 5.2.1. *Let $\mathcal{C}$ be an NF-code based on a number field K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$. Given positive real weights $\beta_1, \ldots, \beta_n$ and any vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$, there are at most l elements $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i)\beta_i \geq \sqrt{\left(\left(1 - \frac{1}{l}\right) d \log(2M/d) + \frac{1}{l}\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i\right)\left(\sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{N}\mathfrak{p}_i}\right)}. \tag{5.3}$$

Proof. Let $\boldsymbol{\alpha} = (\log \mathfrak{N}\mathfrak{p}_1, \ldots, \log \mathfrak{N}\mathfrak{p}_n)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)$ and $(t_1, \ldots, t_n) = (r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n)$. Then

Lemma 4.3.1 implies there exist at most l vectors $(m_1, \ldots, m_n) \in \mathcal{C}$ such that

$$\sum_{j:m_j=t_j} \beta_j \geq \sqrt{\left(\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i - \left(1 - \frac{1}{l}\right) d(\mathcal{C})_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{N}\mathfrak{p}_i}}, \tag{5.4}$$

where $d(\mathcal{C})_{\boldsymbol{\alpha}}$ is the minimum value, over all distinct pairs of vectors $(m_1, \ldots, m_n), (m_1', \ldots, m_n') \in \mathcal{C}$, of the sum $\sum_{j:m\neq m_j'} \alpha_j$.

Let distinct elements $m, m' \in \mathcal{M}_{\mathcal{C}}$ correspond to vectors $(m_1, \ldots, m_n), (m_1', \ldots, m_n') \in \mathcal{C}$. Then the inequality (5.2) implies that

$$\sum_{j:m_j\neq m_j'} \alpha_j = \sum_{i=1}^{n} \big(1 - \sigma(m - m', \mathfrak{p}_i)\big) \log \mathfrak{N}\mathfrak{p}_i \geq \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i - d \log(2M/d).$$

Since m and m' were arbitrary distinct elements of $\mathcal{M}_{\mathcal{C}}$, it follows that $d(\mathcal{C})_{\boldsymbol{\alpha}} \geq \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i - d\log(2M/d)$. Similarly, an element $m \in \mathcal{M}_{\mathcal{C}}$ and its corresponding vector $(m_1, \ldots, m_n) \in \mathcal{C}$ satisfy $\sum_{j:m_j=t_j} \beta_j = \sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i$. Therefore, if $m \in \mathcal{M}_{\mathcal{C}}$ satisfies (5.3), then its corresponding vector $(m_1, \ldots, m_n) \in \mathcal{C}$ satisfies (5.4). Hence, there exist at most l elements $m \in \mathcal{M}_{\mathcal{C}}$ such that (5.3) holds. □

A condition is now derived under which decoding of NF-codes is combinatorially feasible. The proof of the corollary follows that of Corollary 4.3.6, and is therefore omitted. In Section 5.3.4, the condition is used to evaluate the performance of the algorithm for decoding of NF-codes developed in the next section.

Corollary 5.2.2. Let $\mathcal{C}$ be an NF-code based on a number field K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$ such that $\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i \geq d\log(2M/d)$; and $z_1, \ldots, z_n$ be positive real numbers. Then given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$ and any tolerance parameter $\varepsilon > 0$, there are at most polynomially many (in $1/\varepsilon$ and $\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i$) elements $m \in \mathcal{M}_{\mathcal{C}}$ such that

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \sqrt{d\log(2M/d) \left(\sum_{i=1}^{n} z_i^2 \log \mathfrak{N}\mathfrak{p}_i + \varepsilon z_{\max}^2\right)}, \tag{5.5}$$

where $z_{\max} = \max_{1\leq i\leq n} z_i$.

5.3 Weighted List Decoding of NF-codes

In this section, an algorithm for weighted list decoding of NF-codes is developed and analysed. The algorithm's development is based on realising, for the context of NF-codes, the ideal-theoretic framework for list decoding of algebraic error-correcting codes described by Guruswami, Sahai and Sudan [69, Appendix A]. Additionally, the algorithm serves as a natural generalisation of the weighted list decoding algorithm for CRT codes [69] to NF-codes (see Remark 5.3.3). Currently, no method is known for determining shift parameters $\boldsymbol{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ that satisfy (5.1). Therefore, it is assumed that $\boldsymbol{s} = \mathbf{0}$ throughout the section. Decoding in the presence of nonzero shifts is considered in Appendix A.2. Unless stated otherwise, throughout this section $\mathcal{C}$ is an NF-code based on a number field K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$. Furthermore, fix a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$.

A full description and analysis of the framework for list decoding of algebraic error-correcting codes is provided by Guruswami [68, Section 7]. At a high level, the framework suggests the following approach to decoding of NF-codes:

1. Define ideals $I_1, \ldots, I_n \subseteq \mathcal{O}_K[x]$ as follows:

$$I_i = \{\mu(x) \cdot (x - r_i) + \nu(x) \cdot p \mid \mu, \nu \in \mathcal{O}_K[x] \text{ and } p \in \mathfrak{p}_i\}, \quad \text{for } 1 \leq i \leq n.$$

   To each ideal I_i, assign a corresponding positive integer parameter z_i, for $1 \leq i \leq n$.

2. Find a nonzero polynomial $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ of degree at most l and with *small* coefficients.

3. Find the roots of h over K and return all roots in $\mathcal{M}_{\mathcal{C}}$ with sufficiently large weighted agreement.

Realising the approach first requires the development of an explicit notion of a polynomial in $\bigcap_{i=1}^{n} I_i^{z_i}$ with "small" coefficients. For each value of i, $1 \leq i \leq n$, the construction of the ideal I_i implies that a polynomial $h \in I_i$ satisfies the property that $h(m) \in \mathfrak{p}_i^{\sigma(m - r_i, \mathfrak{p}_i)}$, for all $m \in \mathcal{O}_K$. Therefore, for $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ and $m \in \mathcal{O}_K$, the Chinese remainder theorem implies that $h(m) \in \prod_{i=1}^{n} \mathfrak{p}_i^{\sigma(m - r_i, \mathfrak{p}_i) z_i}$. In particular, any $m \in \mathcal{M}_{\mathcal{C}}$ for which the product $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma(m - r_i, \mathfrak{p}_i) z_i}$ is large, is a root of h modulo some ideal of large norm. For $K = \mathbb{Q}$, Howgrave-Graham [78, Section 2] provides a sufficient condition on the coefficient size of $h \in \mathbb{Z}[x]$ for a modular root $m \in \mathbb{Z}$, with $|m| \leq M$, to also be an integer root. A modification of this condition was used by Guruswami et al. in their weighted list decoding algorithm for CRT codes. Here, Howgrave-Graham's condition is generalised to number fields. The condition is then used to obtain an appropriate notion of a polynomial with "small" coefficients.

Define $T_2(x) = \sum_{i=1}^{d} |\sigma_i(x)|^2$, for all $x \in K$. Additionally, given a positive real number M, define $\left\|\sum_i a_i x^i\right\|_{K,M} = \sqrt{\sum_i T_2(a_i) M^{2i}}$, for all $\sum_i a_i x^i \in K[x]$. For polynomials in $\mathcal{O}_K[x]$, the following lemma provides a sufficient condition for a modular root to also be a root over K:

Lemma 5.3.1. *Let K be a number field of degree $[K : \mathbb{Q}] = d$. Let M be a positive real, $\mathfrak{a}$ be a nonzero ideal of $\mathcal{O}_K$, and $h \in \mathcal{O}_K[x]$ be a polynomial of degree at most l. Suppose that*

1. *$h(m) \in \mathfrak{a}$, for some $m \in \mathcal{O}_K$ with $\operatorname{size}(m) \leq M$; and*

2. *$\|h\|_{K,M} < d\,(\mathfrak{N}\mathfrak{a})^{1/d} / \sqrt{l+1}$.*

Then $h(m) = 0$ over K.

Proof. Suppose $h = \sum_{i=0}^{l} h_i x^i \in \mathcal{O}_K[x]$ and $m \in \mathcal{O}_K$ satisfy the conditions of the lemma. Then

$$|N_K(h(m))| \leq \frac{1}{d^d} \text{size}\,(h(m))^d \leq \frac{1}{d^d} \left(\sum_{i=0}^{l} \text{size}\,\left(h_i m^i\right) \right)^d .$$

Applying the Cauchy-Schwarz inequality, it follows that

$$\text{size}(h_i m^i) \leq \sqrt{T_2(h_i) \cdot T_2(m^i)} \leq \sqrt{T_2(h_i) \cdot T_2(m)^i} \leq \sqrt{T_2(h_i) \cdot \text{size}(m)^{2i}}, \quad \text{for } 0 \leq i \leq l.$$

Thus

$$|N_K(h(m))| \leq \frac{1}{d^d} \left(\sum_{i=0}^{l} \sqrt{T_2(h_i) \cdot M^{2i}} \right)^d \leq \frac{1}{d^d} \left(\sqrt{l+1}\, \|h\|_{K,M} \right)^d < \mathfrak{N}\mathfrak{a}.$$

However, $\mathfrak{N}\mathfrak{a}$ divides $N_K(h(m))$, since $h(m) \in \mathfrak{a}$ by assumption. Therefore, $N_K(h(m)) = 0$, otherwise $\mathfrak{N}\mathfrak{a} \leq |N_K(h(m))| < \mathfrak{N}\mathfrak{a}$, which is absurd. Hence, $h(m) = 0$ over K. □

It follows from Lemma 5.3.1 that a polynomial $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ of degree at most l will have among its roots all $m \in \mathcal{M}_{\mathcal{C}}$ such that

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma(m - r_i, \mathfrak{p}_i) z_i} > \frac{1}{d^d} \cdot (l+1)^{\frac{d}{2}} \cdot \|h\|_{K,M}^d \,. \tag{5.6}$$

Motivated by this observation, a polynomial $h \in \mathcal{O}_K[x]$ is said to have small coefficients whenever $\|h\|_{K,M}$ is small. With this notion of size, it follows that two algorithmic tasks arise from the approach to decoding of NF-codes suggested by the general framework: the first, is to find a nonzero polynomial $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ of degree at most l such that $\|h\|_{K,M}$ is small; the second, is to find those roots $m \in \mathcal{M}_{\mathcal{C}}$ of h (over K) with sufficiently large weighted agreement. For the second task, one of several efficient algorithms for factoring polynomials over number fields may be applied (see [14, 150] and references therein), For the first task, algorithms from the geometry of numbers may be employed.

The quadratic form T_2 induces a natural lattice structure for $\mathcal{O}_K$ (see for instance [15]) which can be embedded in $\mathbb{R}^d$ via the so-called *Minkowski map* $\delta_{\mathbb{R}} : K \to \mathbb{R}^d$ defined by mapping $x \in K$ to the vector

$$\left(\sigma_1(x), \ldots, \sigma_{r_1}(x), \sqrt{2}\text{Re}\,(\sigma_{r_1+1}(x))\,, \ldots, \sqrt{2}\text{Re}\,(\sigma_{r_1+r_2}(x))\,, \sqrt{2}\text{Im}\,(\sigma_{r_1+1}(x))\,, \ldots, \sqrt{2}\text{Im}\,(\sigma_{r_1+r_2}(x)) \right).$$

It is readily verified that $T_2(x) = \|\delta_{\mathbb{R}}(x)\|_2^2$ for all $x \in K$, where $\|.\|_2$ is the Euclidean norm on $\mathbb{R}^d$. The Minkowski map is extended to an injective homomorphism from the space of polynomials of degree at most l in $K[x]$ to $\mathbb{R}^{d(l+1)}$ by mapping $\sum_{i=0}^{l} h_i x^i \in K[x]$ to $(\delta_{\mathbb{R}}(h_0), \ldots, \delta_{\mathbb{R}}(h_l))$. Then the embedding in $\mathbb{R}^{d(l+1)}$ of the space of polynomials in $I_i^{z_i}$ with degree at most l forms a lattice. By appropriately scaling this lattice, the problem of finding $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ with $\|h\|_{K,M}$ small is reduced to that of finding a short vector in the resulting lattice.

Using the approach just described, the following algorithm is obtained:

Algorithm 5.3.2.

Input: A code $\mathcal{C}$ based on a number field K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$, where the $\mathfrak{p}_i$ are given in the form $\mathfrak{p}_i = \langle \alpha_i, \beta_i \rangle$ with $\alpha_i \neq 0$, for $1 \leq i \leq n$; a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$; an integral basis $\omega_1, \ldots, \omega_d$ for $\mathcal{O}_K$; and positive integers $z_1, \ldots, z_n$ and l.

Output: All $m \in \mathcal{M}_{\mathcal{C}}$ such that $\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i$ is sufficiently large.

0. For $1 \leq i \leq n$, define the following families of polynomials in $\mathcal{O}_K[x]$:

$$\begin{aligned} a_{i,j,k}(x) &= \alpha_i^{z_i - j}\, \omega_k\, (x - r_i)^j, && \text{for } 0 \leq j \leq \min\{z_i, l\},\ 1 \leq k \leq d; \\ b_{i,j,k}(x) &= \beta_i^{z_i - j}\, \omega_k\, (x - r_i)^j, && \text{for } 0 \leq j \leq \min\{z_i - 1, l\},\ 1 \leq k \leq d; \text{ and} \\ c_{i,j,k}(x) &= \omega_k\, x^j\, (x - r_i)^{z_i}, && \text{for } 1 \leq j \leq l - z_i,\ 1 \leq k \leq d. \end{aligned}$$

1. For $1 \leq i \leq n$, let $L_i \in \mathbb{R}^{d(l+1)}$ be the lattice generated by the vectors $\delta_{\mathbb{R}}(a_{i,j,k})$, $\delta_{\mathbb{R}}(b_{i,j,k})$ and $\delta_{\mathbb{R}}(c_{i,j,k})$. Compute a basis $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d(l+1)})$ for the intersection lattice $L = \bigcap_{i=1}^{n} L_i$.

2. Let $\Omega = \operatorname{diag}(1, \ldots, 1, M, \ldots, M, \ldots, M^l, \ldots, M^l)$, where each power of M occurs d times. Use LLL reduction to find a short vector $\boldsymbol{v}$ in the lattice L_Ω.

3. Recover the polynomial $h = \delta_{\mathbb{R}}^{-1}(\boldsymbol{v}\Omega^{-1})$ and factor it over K.

4. Return all roots $m \in K$ of h such that $m \in \mathcal{M}_{\mathcal{C}}$ and (5.6) holds.

Remark 5.3.3. For $K = \mathbb{Q}$, the ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ are principal: $\mathfrak{p}_i = \langle \gcd(\alpha_i, \beta_i) \rangle$, for $1 \leq i \leq n$. Therefore, it may be assumed that $\beta_i = 0$, for $1 \leq i \leq n$. In this case, Algorithm 5.3.2 reduces to the weighted list decoding algorithm for CRT codes described by Guruswami et al. [69].

5.3.1 Additional Notes on Implementing Algorithm 5.3.2

Algorithm 5.3.2 assumes knowledge of an integral basis for $\mathcal{O}_K$. If an integral basis is not known, finding one currently requires finding the largest squarefree divisor of the discriminant of K (see [139, 27] and references therein). If any of the $\mathfrak{p}_i$ is not given in the form $\mathfrak{p}_i = \langle \alpha_i, \beta_i \rangle$, then such a representation can be found with one of the algorithms described by Cohen [37, Section 1.3] or Belabas [15, Section 6].

The vectors $\delta_{\mathbb{R}}(a_{i,j,k,m})$, $\delta_{\mathbb{R}}(b_{i,j,k,m})$ and $\delta_{\mathbb{R}}(c_{i,j,k})$ may contain non-integer values. In order to avoid floating point arithmetic in Step 1, define a group homomorphism $\delta_{\mathbb{Z}} : K \to \mathbb{Q}^d$ by $a_1\omega_1 + \ldots + a_d\omega_d \mapsto (a_1, \ldots, a_d)$ and extend $\delta_{\mathbb{Z}}$ naturally to a homomorphism from the space of polynomials of degree at most l in $K[x]$ to $\mathbb{Q}^{d(l+1)}$. Then the induced homomorphism $\varphi := \delta_{\mathbb{Z}} \circ \delta_{\mathbb{R}}^{-1}$ permits Step 1 to be performed with operations on integral lattices: if $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{d(l+1)})$ is a basis of $\varphi(L) \subset \mathbb{Z}^{d(l+1)}$, then $(\varphi^{-1}(\boldsymbol{b}_1), \ldots, \varphi^{-1}(\boldsymbol{b}_{d(l+1)}))$ is a basis of L. To compute a basis for the lattice $\varphi(L) = \bigcap_{i=1}^{n} \varphi(L_i)$, note that each lattice $\varphi(L_i)$ is generated by the vectors $\delta_{\mathbb{Z}}(a_{i,j,k})$, $\delta_{\mathbb{Z}}(b_{i,j,k})$ and $\delta_{\mathbb{Z}}(c_{i,j,k})$, for $1 \leq i \leq n$. Therefore, a basis for $\varphi(L_i)$ can be found through Hermite normal form computation [36, Section 2.4.2] or with the MLLL algorithm [140]. Once a basis for each of the lattices $\varphi(L_i)$ has been found, a basis for their intersection $\varphi(L)$ can be computed be repeatedly applying the method described by Guruswami et al. [69, Appendix B].

Computing the vectors $\delta_{\mathbb{Z}}(a_{i,j,k})$, $\delta_{\mathbb{Z}}(b_{i,j,k})$ and $\delta_{\mathbb{Z}}(c_{i,j,k})$ requires computing the coefficients of the polynomials $a_{i,j,k}$, $b_{i,j,k}$ and $c_{i,j,k}$ with respect to the integral basis $\omega_1, \ldots, \omega_d$. In particular, computing the coefficients requires performing polynomially (in d and z_i) many multiplications in K. The reader is referred to Belabas [15], Cohen [36], and Lenstra [107] for further details on arithmetic in number fields. For $x, y \in \mathcal{O}_K$, computing the coefficients of the product xy with respect to $\omega_1, \ldots, \omega_d$ reduces by linearity to computing the coefficients of each of the products $\omega_i \omega_j$. Belabas [15, Proposition 5.1] showed if $(\delta_{\mathbb{R}}(\omega_1), \ldots, \delta_{\mathbb{R}}(\omega_d))$ is LLL-reduced, then the coefficients of the products $\omega_i \omega_j$ will have size polynomial in d and $\log |D_K|$. More general, by modifying Belabas' proof, it can be shown that the coefficients will have size polynomial in d, $\log |D_K|$ and $\sum_{i=1}^{d} T_2(\omega_i)$. Therefore, given the products $\omega_i \omega_j$, it is possible to compute the coefficients of the polynomials $a_{i,j,k}$, $b_{i,j,k}$ and $c_{i,j,k}$ with respect to $\omega_1, \ldots, \omega_d$ in time polynomial in d, z_i, $\log \|\delta_{\mathbb{Z}}(\alpha_i)\|_2$, $\log \|\delta_{\mathbb{Z}}(\beta_i)\|_2$, $\sum_{i=1}^{d} \log T_2(\omega_i)$ and $\log |D_K|$.

If K is totally real and $M \in \mathbb{Q}$, then the lattice reduction in Step 2 may be performed with an integral reduction algorithm [36, 87] or a floating point variant such as the L^2 algorithm [132, 131]. Otherwise, lattice reduction is required to be performed on real-valued bases for which far less is known about the stability and complexity of the LLL algorithm. In Step 2, the requirement that a reduced basis for L_Ω is found may be replaced with the requirement that a short vector in L_Ω is found. Consequently, a method described by Belabas [15, Section 4.2], which uses integral reduction to produce a basis that contains a short vector, may be used in Step 2 in place of reduction on real-valued bases. In Section 5.3.3, an idea of Fieker and Friedrichs [59] is used to modify Algorithm 5.3.2 such that integral lattice reduction can be used for non-totally real fields (whenever M is rational). However, the decoding performance of the resulting algorithm does not match that of Algorithm 5.3.2.

Depending on the method of reduction used, it may only be possible to find a floating point approximation to a short vector in L_Ω. Assuming the approximation has been calculated with sufficient precision, it is possible to recover the coefficients of the polynomial h with one of the methods described by Belabas [15, Section 3.2].

5.3.2 Analysis of the Decoding Algorithm

In this section, the decoding performance of Algorithm 5.3.2 is analysed. To begin, the space of polynomials corresponding to vectors in L_i is shown to be exactly the polynomials of degree at most l in the ideal $I_i^{z_i}$. It follows immediately that the space of polynomials corresponding to vectors in the intersection lattice $L = \bigcap_{i=1}^{n} L_i$ is exactly the polynomials of degree at most l in the ideal $\bigcap_{i=1}^{n} I_i^{z_i}$.

Lemma 5.3.4. For each value of i, $1 \leq i \leq n$, the space of polynomials corresponding to vectors in L_i is exactly the polynomials of degree at most l in the ideal $I_i^{z_i}$.

Proof. Fix a value of i, $1 \leq i \leq n$. By construction, L_i is the space of all $\mathbb{Z}$-linear combinations of the vectors $\delta_{\mathbb{R}}(a_{i,j,k})$, $\delta_{\mathbb{R}}(b_{i,j,k})$ and $\delta_{\mathbb{R}}(c_{i,j,k})$. Moreover, each of the polynomials $a_{i,j,k}$, $b_{i,j,k}$ and $c_{i,j,k}$ clearly belongs to the ideal $I_i^{z_i}$ and has degree at most l. Therefore, the lemma will follow by showing that those polynomials of degree at most l in the ideal $I_i^{z_i}$ can each be expressed as a $\mathbb{Z}$-linear combination of the polynomials $a_{i,j,k}$, $b_{i,j,k}$ and $c_{i,j,k}$.

Given a polynomial $h \in I_i^{z_i}$ such that $\deg h \leq l$, the definition of ideal I_i implies that h may be written in the form

$$h(x) = \lambda_{z_i}(x) \cdot (x - r_i)^{z_i} + \sum_{j=0}^{z_i - 1} \lambda_j(x) \cdot p_j (x - r_i)^j,$$

where $p_j \in \mathfrak{p}_i^{z_i - j}$, for $1 \leq j \leq z_i - 1$; and $\lambda_0, \ldots, \lambda_{z_0} \in \mathcal{O}_K[x]$. Moreover, since $(x - r_i) \cdot p_j(x - r_i)^j = p_j \cdot (x - r_i)^{j+1}$ and $\mathfrak{p}_i^{z_i - j} \subset \mathfrak{p}_i^{z_i - j - 1}$, for $0 \leq j \leq z_i - 1$, it may be assumed that $\lambda_0, \ldots, \lambda_{z_i - 1} \in \mathcal{O}_K$. It follows that $\lambda_j = 0$, for all $0 \leq j \leq z_i$ such that $l < j$.

For any integer $t \geq 0$, $\mathfrak{p}_i^t = \langle \alpha_i^t, \beta_i^t \rangle$. Therefore, for all $0 \leq j \leq z_i - 1$ such that λ_j is nonzero, $\lambda_j \cdot p_j(x - r_i)^j$ may be expressed as a $\mathbb{Z}$-linear combination of the polynomials $a_{i,j,k}$ and $b_{i,j,k}$, for $0 \leq j \leq \min\{z_i - 1, l\}$, $1 \leq k \leq d$. Similarly, if λ_{z_i} is nonzero, then $\lambda_{z_i}(x) \cdot (x - r_i)^{z_i}$ may be expressed as a $\mathbb{Z}$-linear combination of the polynomials $a_{i,z_i,k}$ and $c_{i,j,k}$, for $1 \leq j \leq l - z_i$, $1 \leq k \leq d$. □

The following lemma provides an upper bound on the determinant of the scaled lattice L_Ω. Combining the bound with Theorem 3.1.2 provides a bound on $\|h\|_{K,M}$, where h is the polynomial found in Step 3 of Algorithm 5.3.2, thus enabling the decoding performance of Algorithm 5.3.2 to be determined.

Lemma 5.3.5. The lattice L_Ω is $d(l+1)$-dimensional and

$$\det L_\Omega \leq M^{d\binom{l+1}{2}} \cdot |D_K|^{\frac{l+1}{2}} \cdot \prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}}.$$

Proof. Let Λ and Λ_0 be the space of polynomials of degree at most l in $\mathcal{O}_K[x]$ and $I = \bigcap_{i=1}^n I_i^{z_i}$ respectively. Let A be the $d \times d$ matrix whose rows are the vectors $\delta_{\mathbb{R}}(\omega_1), \ldots, \delta_{\mathbb{R}}(\omega_d)$. Then $|\det A| = |\det(\sigma_i(\omega_j))| = |D_K|^{1/2}$. Let B be the $d(l+1) \times d(l+1)$ block diagonal matrix with each block on the diagonal equal to A. The row vectors of B are linearly independent since $|\det B| = |D_K|^{(l+1)/2} \neq 0$ and their span is clearly equal to $\delta_{\mathbb{R}}(\Lambda)$. Hence $\delta_{\mathbb{R}}(\Lambda)$ is a $d(l+1)$-dimensional lattice and $\det \delta_{\mathbb{R}}(\Lambda) = |D_K|^{(l+1)/2}$.

Define $\pi : \Lambda \to \mathcal{O}_K[x]/I$ by $\lambda \mapsto \lambda + I$. Then π is a homomorphism of abelian groups and $\ker \pi = \Lambda_0$. In addition, $|\mathcal{O}_K[x]/I| \leq \prod_{i=1}^n \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}}$ (see [68, Corollary 7.5]). Therefore, $|\mathcal{O}_K[x]/I|$ is finite and $|\Lambda/\Lambda_0|$ must divide $|\mathcal{O}_K[x]/I|$. Since $\delta_{\mathbb{R}}$ is an injective homomorphism, it follows that $|\Lambda/\Lambda_0| = [\delta_{\mathbb{R}}(\Lambda) :$

$\delta_{\mathbb{R}}(\Lambda_0)]$. Hence, $\delta_{\mathbb{R}}(\Lambda_0)$ is a full-rank sublattice of $\delta_{\mathbb{R}}(\Lambda)$ and

$$\det \delta_{\mathbb{R}}(\Lambda_0) = \det \delta_{\mathbb{R}}(\Lambda) \cdot [\delta_{\mathbb{R}}(\Lambda) : \delta_{\mathbb{R}}(\Lambda_0)] \leq |D_K|^{\frac{l+1}{2}} \cdot |\mathcal{O}_K[x]/I|.$$

It follows from Lemma 5.3.4 that $L = \delta_{\mathbb{R}}(\Lambda_0)$. Therefore, L is a $d(l+1)$-dimensional lattice and

$$\det L \leq |D_K|^{\frac{l+1}{2}} \cdot \prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}}.$$

The proof is completed by noting that $|\det \Omega| = M^{d\binom{l+1}{2}}$ is nonzero, thus L_Ω is a $d(l+1)$-dimensional lattice and $\det L_\Omega = \det L \cdot |\det \Omega|$. □

The following theorem provides a sufficient condition for an element $m \in \mathcal{M}_{\mathcal{C}}$ to be returned by Algorithm 5.3.2:

Theorem 5.3.6. *Algorithm 5.3.2 returns all $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma(m-r_i,\mathfrak{p}_i)z_i} > 2^{\frac{d^2(l+1)-d}{4}} d^{-d} (l+1)^{\frac{d}{2}} M^{\frac{dl}{2}} \sqrt{|D_K|} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}} \right)^{\frac{1}{l+1}}. \tag{5.7}$$

Proof. If $\boldsymbol{v} \in L_\Omega$ is the vector returned in Step 2 of Algorithm 5.3.2 and $h = \delta_{\mathbb{R}}^{-1}(\boldsymbol{v}\Omega^{-1})$ the corresponding polynomial, then $\|h\|_{K,M} = \|\boldsymbol{v}\|_2$. Moreover, it follows from Theorem 3.1.2 that $\boldsymbol{v}$ satisfies

$$\|\boldsymbol{v}\|_2 \leq 2^{\frac{d(l+1)-1}{4}} \det(L_\Omega)^{\frac{1}{d(l+1)}}.$$

This inequality, Lemma 5.3.4 and Lemma 5.3.5 imply that $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ and

$$\|h\|_{K,M} \leq 2^{\frac{d(l+1)-1}{4}} M^{\frac{l}{2}} |D_K|^{\frac{1}{2d}} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}} \right)^{\frac{1}{d(l+1)}}.$$

Therefore, given $m \in \mathcal{M}_{\mathcal{C}}$ such that (5.7) holds, applying Lemma 5.3.1 with $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{p}_i^{\sigma(m-r_i,\mathfrak{p}_i)z_i}$ implies that $h(m) = 0$ over K. Hence, all such $m \in \mathcal{M}_{\mathcal{C}}$ are found in Step 4 of the algorithm. □

Remark 5.3.7. Given a monic polynomial $f \in \mathcal{O}_K[x]$, a nonzero ideal $\mathfrak{a} \subset \mathcal{O}_K$, a positive real number M, and positive integers k, l such that $l \geq k \deg f$, the polynomial time algorithm described by Cohn and Heninger [38] in the proof of their Theorem 1.3, when applied with parameters f, $I = \mathfrak{a}$, $\lambda_1 = \ldots = \lambda_d = M$, k and $t = l + 1 - k \deg f$, returns all $m \in \mathcal{O}_K$ such that $\operatorname{size}(m) \leq M$ and

$$\mathfrak{N}\gcd\left((f(m)), \mathfrak{a}\right)^k > \left(2^{\frac{d(l+1)-1}{4}} \sqrt{\frac{l+1}{d}} |D_K|^{\frac{1}{2d}} M^{\frac{l}{2}} \right)^d \left(\mathfrak{N}\mathfrak{a}^{\binom{k+1}{2}} \right)^{\frac{\deg f}{l+1}}.$$

Given an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$, positive integer weights $z_1, \ldots, z_n$, and a received word $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{p}_n$, weighted list decoding is performed with their algorithm by additionally setting $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{p}_i$, $k = \max_{1 \le i \le n} z_i$ and $f = x - r$, where $r \in \mathcal{O}_K$ satisfies $r - r_i \in \mathfrak{p}_i$, for $1 \le i \le n$. When the weights $z_1, \dots, z_n$ are equal, this approach performs almost identically to Algorithm 5.3.2, with both methods providing a generalisation of the list decoding algorithm for CRT codes of Boneh [25].

5.3.3 Decoding with Integral Lattices

Let $\omega_1, \dots, \omega_d$ be an integral basis for $\mathcal{O}_K$. For $x = \sum_{i=1}^{d} x_i \omega_i \in K$, the Cauchy-Schwarz inequality implies that

$$T_2(x) \le \left(\sum_{i=1}^{d} x_i^2 \right) \left(\sum_{i=1}^{d} T_2(\omega_i) \right) = \|\delta_{\mathbb{Z}}(x)\|_2^2 \cdot \sum_{i=1}^{d} T_2(\omega_i). \tag{5.8}$$

It follows that $T_2(x)$ is small whenever both $\|\delta_{\mathbb{Z}}(x)\|_2$ and $\sum_{i=1}^{d} T_2(\omega_i)$ are small. Recall the homomorphism $\varphi = \delta_{\mathbb{Z}} \circ \delta_{\mathbb{R}}^{-1}$ defined in Section 5.3.1. Given an integral basis $\omega_1, \dots, \omega_d$ with the property that $\sum_{i=1}^{d} T_2(\omega_i)$ is small, (5.8) implies that performing lattice reduction on $\varphi(L)_\Omega$ instead of L_Ω in Algorithm 5.3.2 will still produce a polynomial $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ such that $\|h\|_{K,M}$ is small. Performing lattice reduction on $\varphi(L)_\Omega$ is beneficial as it permits the use of an integral lattice reduction algorithm whenever M is rational, regardless of whether K is totally real. Furthermore, the use of exact arithmetic saves the reconstruction of algebraic numbers from approximations to their embeddings. This simplified approach is based on an idea due to Fieker and Friedrichs [59] which led to a considerable speed-up for non-totally real fields in their application.

Theorem 5.3.8. *There exists a positive constant C_1 that depends only on the input basis $\omega_1, \dots, \omega_d$ such that Algorithm 5.3.2, with LLL performed on $\varphi(L)_\Omega$, outputs all $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma(m - r_i, \mathfrak{p}_i) z_i} > 2^{\frac{d^2(l+1)-d}{4}} d^{-d} C_1^{\frac{d}{2}} (l+1)^{\frac{d}{2}} M^{\frac{dl}{2}} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}} \right)^{\frac{1}{l+1}}. \tag{5.9}$$

Proof. Let A be the $d \times d$ matrix whose rows are the vectors $\delta_{\mathbb{R}}(\omega_1), \dots, \delta_{\mathbb{R}}(\omega_d)$. Then $\delta_{\mathbb{R}}(x) = \delta_{\mathbb{Z}}(x) \cdot A$ for all $x \in K$. More generally, if B is the $d(l+1) \times d(l+1)$ block diagonal matrix with each block on the diagonal equal to A, then $\delta_{\mathbb{R}}(b) = \delta_{\mathbb{Z}}(b) \cdot B$, for all $b \in \mathcal{O}_K[x]$ with $\deg b \le l$. Therefore, $\varphi(L)$ is a $d(l+1)$-dimensional lattice and $\det \varphi(L) = |\det B|^{-1} \cdot \det L$. Hence,

$$\det \varphi(L)_\Omega = |\det B|^{-1} \cdot |\det \Omega| \cdot \det L = |D_K|^{-\frac{l+1}{2}} \cdot \det L_\Omega.$$

Let C_1 be the smallest real number such that $T_2(x) \le C_1 \|\delta_{\mathbb{Z}}(x)\|_2^2$ for all $x \in K$, which exists and is positive by (5.8). Therefore, if $\boldsymbol{v} \in \varphi(L)_\Omega$ is the shortest basis vector returned by performing LLL reduction on $\varphi(L)_\Omega$, then the corresponding polynomial $h = \delta_{\mathbb{Z}}^{-1}(\boldsymbol{v}\Omega^{-1})$ satisfies $\|h\|_{K,M} \le \sqrt{C_1} \|\boldsymbol{v}\|_2$. Moreover, it follows from Theorem 3.1.2 that $\boldsymbol{v}$ satisfies

$$\|\boldsymbol{v}\|_2 \le 2^{\frac{d(l+1)-1}{4}} \det(\varphi(L)_\Omega)^{\frac{1}{d(l+1)}}.$$

This inequality, Lemma 5.3.4 and Lemma 5.3.5 imply that $h \in \bigcap_{i=1}^{n} I_i^{z_i}$ and

$$\|h\|_{K,M} \leq 2^{\frac{d(l+1)-1}{4}} C_1^{\frac{1}{2}} M^{\frac{l}{2}} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}} \right)^{\frac{1}{d(l+1)}}.$$

Therefore, given $m \in \mathcal{M}_{\mathcal{C}}$ such that (5.9) holds, applying Lemma 5.3.1 with $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{p}_i^{\sigma(m-r_i,\mathfrak{p}_i)z_i}$ implies that $h(m) = 0$ over K. □

As noted by Fieker and Friedrichs [59], a standard result from numerical analysis states that the constant C_1 is equal to the largest eigenvalue of AA^t, where A is the matrix whose rows are the vectors $\delta_{\mathbb{R}}(\omega_1), \ldots, \delta_{\mathbb{R}}(\omega_d)$ (see [158, Section 4.4]). Let $\lambda_1, \ldots, \lambda_d$ be the eigenvalues (with multiplicity) of AA^t, which are real and positive since AA^t is positive definite Hermitian. Since the determinant of a matrix is equal to the product of its eigenvalues, an application of the AM-GM inequality shows that

$$C_1 = \max_{1 \leq i \leq d} \lambda_i \geq \frac{1}{d}(\lambda_1 + \ldots + \lambda_d) \geq (\lambda_1 \cdots \lambda_d)^{\frac{1}{d}} = \left(\det AA^t\right)^{\frac{1}{d}} = |D_K|^{\frac{1}{d}}.$$

Hence, for all possible choices of integral basis $\omega_1, \ldots, \omega_d$, the decoding condition (5.9) will never surpass (5.7). If $\omega_1, \ldots, \omega_d$ has the property that $(\delta_{\mathbb{R}}(\omega_1), \ldots, \delta_{\mathbb{R}}(\omega_d))$ is LLL-reduced, then for any other basis $\omega_1', \ldots, \omega_d'$ of $\mathcal{O}_K$, Theorem 3.1.2 implies that

$$T_2(\omega_i) \leq 2^{d-1} \cdot \max_{1 \leq j \leq d} T_2(\omega_j'), \quad \text{for } 1 \leq i \leq d.$$

Thus choosing another basis for $\mathcal{O}_K$ may only reduce C_1 by a factor of at most $d2^{d-1}$. Therefore, when using Algorithm 5.3.2 with lattice reduction performed on $\varphi(L)_\Omega$, it is beneficial to choose a basis $\omega_1, \ldots, \omega_d$ such that $(\delta_{\mathbb{R}}(\omega_1), \ldots, \delta_{\mathbb{R}}(\omega_d))$ is LLL-reduced.

5.3.4 Parameter Selection for Algorithm 5.3.2

To begin this section, Theorem 5.3.6 is modified to permit arbitrary (positive) real parameters $z_1, \ldots, z_n$. Then the performance of Algorithm 5.3.2 is evaluated against the theoretical bounds obtained in Section 5.2. Finally, the performance of Algorithm 5.3.2 as a method for performing weighted and traditional list decoding is considered. The results of this section are straightforward generalisations of results obtained by Guruswami, Sahai and Sudan [69, Section 3.4] for decoding of CRT codes (see also [68, Section 7.6.3]). Consequently, their proofs are either abridged or omitted entirely.

Theorem 5.3.9. *Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$. For nonnegative reals $z_1, \ldots, z_n$ and any tolerance parameter $\varepsilon > 0$, given a* *vector* $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$*, Algorithm 5.3.2 can list all* $m \in \mathcal{M}_\mathcal{C}$ *such that*

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \sqrt{d \log\left(2^{\frac{d}{2}} M\right) \left(\sum_{i=1}^{n} z_i^2 \log \mathfrak{N}\mathfrak{p}_i + \varepsilon z_{\max}^2\right)}, \tag{5.10}$$

where $z_{\max} = \max_{1 \leq i \leq n} z_i$*.*

Proof. It can be assumed that the z_i's are all nonzero. Moreover, since (5.10) is invariant under scaling of the z_i's, it can be assumed without loss of generality that $z_{\max} = 1$. Let A be an integer parameter to be determined later and set $z_i^* = \lceil A z_i \rceil$, for $1 \leq i \leq n$. Consequently, $A z_i \leq z_i^* \leq A z_i + 1$, for $1 \leq i \leq n$. Therefore, Theorem 5.3.6 implies that on the input of parameters $z_1^*, \ldots, z_n^*$ and a positive integer l, Algorithm 5.3.2 can list all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \frac{1}{2A}\left(dl \log(2^{\frac{d}{2}} M) + d \log(l+1)\right)$$
$$+ \frac{d}{A} \log\left(2^{\frac{d-1}{4}} d^{-1} |D_K|^{\frac{1}{2d}}\right) + \frac{A}{2(l+1)} \sum_{i=1}^{n} \left(z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2}\right) \log \mathfrak{N}\mathfrak{p}_i. \tag{5.11}$$

Define $Z_i = z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2}$, for $1 \leq i \leq n$, and set

$$l = \left\lceil A \sqrt{\frac{\sum_{i=1}^{n} Z_i \log \mathfrak{N}\mathfrak{p}_i}{d \log(2^{\frac{d}{2}} M)}} \right\rceil - 1.$$

Then (5.11) is satisfied by all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \frac{d}{2A} \log\left(A \sqrt{\frac{\sum_{i=1}^{n} Z_i \log \mathfrak{N}\mathfrak{p}_i}{d \log(2^{\frac{d}{2}} M)}} + 1\right)$$
$$+ \frac{d}{A} \log\left(2^{\frac{d-1}{4}} d^{-1} |D_K|^{\frac{1}{2d}}\right) + \sqrt{d \log(2^{\frac{d}{2}} M) \sum_{i=1}^{n} Z_i \log \mathfrak{N}\mathfrak{p}_i}.$$

There exists a positive constant A_0 such that, for all $A \geq A_0$, the right hand side of the inequality is less than or equal to the right hand side of (5.10). The proof is completed by setting $A = A_0$. □

A tedious calculation shows that the constant A_0 in the proof of Theorem 5.3.9 has size which is at most polynomial in d, $\log |D_K|$, $\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i$ and $1/\varepsilon$. As a result, for the parameters used in the proof, Algorithm 5.3.2 performs lattice reduction on a lattice of dimension polynomial in d, $\log |D_K|$, $\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i$ and $1/\varepsilon$. If lattice reduction is performed on $\varphi(L)_\Omega$, as proposed in Section 5.3.3, then only superficial modifications are required to show that (5.10) can still be obtained with reduction performed on a lattice of dimension polynomial in d, $\log C_1$, $\sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{p}_i$ and $1/\varepsilon$. Comparing conditions (5.10) to (5.5) suggests that Algorithm 5.3.2 does not perform optimally for $d \neq 1$.

Theorem 5.3.9 provides a condition under which Algorithm 5.3.2 can be used to find all $m \in \mathcal{M}_\mathcal{C}$ that satisfy a certain "weighted" condition. However, weighted list decoding requires that, for given positive real weights $\beta_1, \ldots, \beta_n$ and some $t \geq 0$, all $m \in \mathcal{M}_\mathcal{C}$ such that $\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i)\beta_i \geq t$ are found. The following corollary to Theorem 5.3.9 determines how small t can be taken when using Algorithm 5.3.2, thus describing the algorithms performance as a method for performing weighted list decoding:

Corollary 5.3.10. Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$. For nonnegative weights $\beta_1, \ldots, \beta_n$ and any tolerance parameter $\varepsilon > 0$, given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$, Algorithm 5.3.2 can list all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i)\beta_i \geq \sqrt{d \log\left(2^{\frac{d}{2}} M\right)\left(\sum_{i=1}^n \beta_i^2 + \varepsilon \max_{1 \leq i \leq n} \frac{\beta_i^2}{\log^2 \mathfrak{N}\mathfrak{p}_i}\right)}.$$

To end the section, two additional corollaries to Theorem 5.3.9, which describe the performance of Algorithm 5.3.2 as a method for performing (traditional) list decoding, are now provided:

Corollary 5.3.11. Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$. Given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$ and any tolerance parameter $\varepsilon > 0$, Algorithm 5.3.2 can list all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i) \geq \sqrt{d \log\left(2^{\frac{d}{2}} M\right)\left(\sum_{i=1}^n \frac{1}{\log \mathfrak{N}\mathfrak{p}_i} + \varepsilon\right)}.$$

Corollary 5.3.12. Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \mathbf{0})$. Suppose that $\prod_{i=1}^n \mathfrak{N}\mathfrak{p}_i \geq 2^{d^2/2} M^d$ and let k be the least integer such that $\prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i \geq 2^{d^2/2} M^d$. Then given a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$ and any tolerance parameter $\varepsilon > 0$, Algorithm 5.3.2 can list all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i) \geq \sqrt{k(n + 1 + \varepsilon)}.$$

If $\prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i = 2^{d^2/2} M^d$, the bound becomes $\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i) \geq \sqrt{k(n + \varepsilon)}$.

Proof. Let k be the least integer such that $\prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i \geq 2^{d^2/2} M^d$. Define $z_i = 1/\log \mathfrak{N}\mathfrak{p}_k$, for $1 \leq i \leq k$; and $z_i = 1/\log \mathfrak{N}\mathfrak{p}_i$, for $k < i \leq n$. Then

$$\sum_{i=1}^n \sigma(x, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i = \sum_{i=1}^k \sigma(x, \mathfrak{p}_i)\left(\frac{\log \mathfrak{N}\mathfrak{p}_i}{\log \mathfrak{N}\mathfrak{p}_k} - 1\right) + \sum_{i=1}^n \sigma(x, \mathfrak{p}_i), \quad \text{for all } x \in \mathcal{O}_K.$$

Moreover, $\frac{\log \mathfrak{N}\mathfrak{p}_i}{\log \mathfrak{N}\mathfrak{p}_k} - 1 \leq 0$, for $1 \leq i \leq k$. Consequently, for $x \in \mathcal{O}_K$,

$$\sum_{i=1}^{n} \sigma(x, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \sum_{i=1}^{k} \left(\frac{\log \mathfrak{N}\mathfrak{p}_i}{\log \mathfrak{N}\mathfrak{p}_k} - 1 \right) + \sum_{i=1}^{n} \sigma(x, \mathfrak{p}_i) \geq \frac{d \log \left(2^{d/2} M\right)}{\log \mathfrak{N}\mathfrak{p}_k} - k + \sum_{i=1}^{n} \sigma(x, \mathfrak{p}_i).$$

Applying Theorem 5.3.9 with tolerance parameter $\varepsilon' = \varepsilon \log \mathfrak{N}\mathfrak{p}_k$, it follows that Algorithm 5.3.2 can list all $m \in \mathcal{M}_{\mathcal{C}}$ such that

$$\sum_{i=1}^{n} \sigma(m - r_i, \mathfrak{p}_i) \geq k - \frac{d \log \left(2^{d/2} M\right)}{\log \mathfrak{N}\mathfrak{p}_k} + \sqrt{\frac{d \log \left(2^{d/2} M\right)}{\log \mathfrak{N}\mathfrak{p}_k} \left(\frac{d \log \left(2^{d/2} M\right)}{\log \mathfrak{N}\mathfrak{p}_k} + \sum_{i=k}^{n} \frac{\log \mathfrak{N}\mathfrak{p}_k}{\log \mathfrak{N}\mathfrak{p}_i} + \varepsilon \right)}.$$

If $\prod_{i=1}^{k} \mathfrak{N}\mathfrak{p}_i = 2^{d^2/2} M^d$, then the lower index of the sum on the left hand side can be changed to $k+1$. In either case, the remainder of the proof follows that of Guruswami et al. [69, Theorem 5]. □

5.4 Smooth Algebraic Integers in Number Fields

An integer x is called *y-smooth* if it is free of prime factors greater than y. Smooth numbers play an important role in many algorithms from computational number theory and cryptography (see surveys by Granville [66] and Pomerance [146]). Boneh [25] demonstrated how to use a list decoding algorithm for CRT codes to search short intervals for integers containing a large smooth factor. Furthermore, Boneh showed that the list size of the corresponding instances of the CRT decoding problem provide an upper bound on the number of integers with large smooth factors. In this section, Boneh's results are generalised to number fields by replacing the CRT decoding algorithm with the algorithm for NF-codes described in Section 5.3. To begin, the notion of a smooth integer is generalised to number fields:

Definition 5.4.1. Let K be a number field and $y > 0$ be an integer. Then an element $x \in \mathcal{O}_K$ is called *y-smooth* if $N_K(x)$ is a y-smooth integer. Furthermore, an element $x \in \mathcal{O}_K$ is said to have a *y-smooth factor* whenever $N_K(x)$ has a y-smooth factor.

An element $x \in \mathcal{O}_K$ is y-smooth if and only if for every prime ideal $\mathfrak{p} \subset \mathcal{O}_K$, $\mathfrak{p} \mid \langle x \rangle$ implies that $\mathfrak{p}$ lies over a rational prime $p \leq y$. Therefore, the smoothness of an element $x \in \mathcal{O}_K$ describes how the principal ideal it generates factorises over small prime ideals. It is natural to consider the factorisation into ideals since $\mathcal{O}_K$ may fail to have unique factorisation. Algebraic integers in number fields that are smooth, or have a smooth factor, appear in many number-theoretic algorithms. These algorithms include, but are not limited to, algorithms for integer factorisation [102], discrete logarithms in finite fields [64, 86], finding solutions to the Pell equation [109] and computing class groups [28].

5.4.1 Finding Smooth Algebraic Integers in Number Fields

Boneh's algorithm searches for smooth integers in an interval $[U, V]$ with length determined by the number of errors correctable by a CRT list decoding algorithm. For $K \neq \mathbb{Q}$, an analogous problem is to search for smooth algebraic integers in some *ball* defined by the size function $\text{size}(x)$:

Definition 5.4.2. Let $c \in \mathcal{O}_K$ and M be a nonnegative real number. Then the *ball of radius M centred at c* is define to be $B_{c,M} = \{x \in \mathcal{O}_K \mid \text{size}(x - c) \leq M\}$.

For $K = \mathbb{Q}$, the ball of radius M centred at $c \in \mathbb{Z}$ is simply the interval $B_{c,M} = [c - M, c + M]$. The method put forward by Boneh for utilising CRT decoding to find smooth integers in such an interval is now briefly summarised. Let $\mathcal{C}$ be the code based on $\mathbb{Q}$ with parameters $(n, \langle p_1^{w_1} \rangle, \ldots, \langle p_n^{w_n} \rangle; M, \mathbf{0})$, where $p_1, \ldots, p_n$ are the primes up to y; and $w_1, \ldots, w_n$ are positive integers. Applying Algorithm 5.3.2 to $\mathcal{C}$ and $\boldsymbol{r} = (-c + \langle p_1^{w_1} \rangle, \ldots, -c + \langle p_n^{w_n} \rangle)$ returns all integers $m \in [-M, M]$ such that $\sigma(m + c, \langle p_i^{w_i} \rangle) = 1$ for many values of i, $1 \leq i \leq n$. For each such m, the integer $x = m + c$ satisfies $x \in [c - M, c + M]$ and $p_i^{w_i} \mid x$ for many values of i, $1 \leq i \leq n$. This approach does not account for higher powers of p_i that possibly divide $x \in B_{c,M}$. Therefore, some integers that contain a y-smooth factor may be missed. However, the following lemma implies that the approach does account for the contribution to the factorisation of $x \in B_{c,M}$ made by all prime powers p_i^k such that $0 \leq k \leq w_i$:

Lemma 5.4.3. For an ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, an element $r \in \mathcal{O}_K$, and a positive integer w, define

$$I = \{\mu(x) \cdot (x - r) + \nu(x) \cdot a \mid \mu, \nu \in \mathcal{O}_K[x] \text{ and } a \in \mathfrak{a}^w\}.$$

Suppose that $c \in I^z$, for some integer $z \geq 1$. Then, given $m \in \mathcal{O}_K$, if σ is the largest integer such that $0 \leq \sigma \leq w$ and $m - r \in \mathfrak{a}^\sigma$, it follows that $c(m) \in \mathfrak{a}^{\sigma z}$.

Proof. Let $m \in \mathcal{O}_K$, and define σ to be the largest integer such that $0 \leq \sigma \leq w$ and $m - r \in \mathfrak{a}^\sigma$. For all $b \in I$, it is clear that $b(m) \in \mathfrak{a}^\sigma$. Therefore, given an integer $z \geq 1$ and a polynomial $c \in I^z$, it follows that $c(m) \in \mathfrak{a}^{\sigma z}$. □

Throughout the remainder of the chapter, the following notation is used: given an ideal $\mathfrak{a} \subseteq \mathcal{O}_K$ and a positive integer w, define $\sigma_w^*(x, \mathfrak{a})$, for $x \in \mathcal{O}_K$, to be the largest integer σ such that $0 \leq \sigma \leq w$ and $x \in \mathfrak{a}^\sigma$. Using this notation, the main algorithmic result of this section may be stated as follows:

Theorem 5.4.4. *Let K be number field with $[K : \mathbb{Q}] = d$ that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$. Let $w_1, \ldots, w_n, z_1, \ldots, z_n, l$ be positive integers and M be a positive real number. Then for all $c \in \mathcal{O}_K$, Algorithm 5.3.2 can be used to find all $x \in B_{c,M}$ such that*

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x, \mathfrak{p}_i) z_i} > 2^{\frac{d^2(l+1)-d}{4}} \, d^{-d} \, (l+1)^{\frac{d}{2}} \, M^{\frac{dl}{2}} \, \sqrt{|D_K|} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{w_i \binom{z_i+1}{2}} \right)^{\frac{1}{l+1}}. \tag{5.12}$$

Proof. Let $\mathcal{C}$ be the NF-code based on K with parameters $(n, \mathfrak{p}_1^{w_1}, \ldots, \mathfrak{p}_n^{w_n}; M, \mathbf{0})$. Then

$$B_{c,M} = \{x \in \mathcal{O}_K \mid \text{size}(x - c) \leq M\} = \{m + c \mid m \in \mathcal{O}_K \text{ and size}(m) \leq M\} = \{m + c \mid m \in \mathcal{M}_{\mathcal{C}}\}.$$

By using Lemma 5.4.3 and adapting the proof of Theorem 5.3.6 accordingly, it follows that Algorithm 5.3.2, when applied to $\mathcal{C}$ and vector $(-c + \mathfrak{p}_1, \ldots, -c + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{p}_n$, returns all $m \in \mathcal{M}_{\mathcal{C}}$ such that $x = m + c$ satisfies the conditions of the theorem. □

Taking $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ to consist of all prime ideals in $\mathcal{O}_K$ such that $\mathfrak{N}\mathfrak{p}_i \leq y$, and setting $z_1 = \cdots = z_n$, the algorithm described by Theorem 5.4.4 can be used to find $x \in B_{c,M}$ such that $N_K(x)$ has a large y-smooth factor. It should be noted that this method does not necessarily generate y-smooth elements. Of course, an output x to the algorithm will be smooth whenever the smooth factor of $N_K(x)$ is equal to $N_K(x)$ itself.

Remark 5.4.5. Boneh [25, Section 4] presented a generalisation of his CRT list decoding algorithm that takes a low degree polynomial $f \in \mathbb{Z}[x]$ and returns integers $x \in [U, V]$ such that $f(x)$ contains a large smooth factor. It is possible to generalise Boneh's approach to NF-codes in order to obtain an algorithm for finding $x \in \mathcal{O}_K$ of bounded size such that $\mathfrak{N}\langle f(x)\rangle$ contains a large smooth factor. Here the polynomial f now belongs to $\mathcal{O}_K[x]$. As discussed in Remark 5.3.7, such an algorithm has been obtained by Cohn and Heninger [38]. Additionally, an analogous problem inside rings of the form $\mathbb{Z}[\alpha]$, where α is an algebraic number, was previously considered by Howgrave-Graham [79, Section 4.7].

In practice, it is often beneficial to choose the dimension of the lattice occurring in Algorithm 5.3.2 to suit the specific implementation of lattice reduction and available computational recourses. The following corollary to Theorem 5.4.3 considers parameter selection when l is fixed:

Corollary 5.4.6. Let K be a number field with $[K : \mathbb{Q}] = d$ that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$. Let $w_1, \ldots, w_n, l$ be positive integers and $M \geq d$ be a real number. Then for any $c \in \mathcal{O}_K$, Algorithm 5.3.2 can be used to find all $x \in B_{c,M}$ such that

$$\sum_{i=1}^{n} \sigma_{w_i}^*(x, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i \geq \sqrt{\frac{dl}{l+1}\left(\log M + \log \eta(d, l)\right) \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i} + \frac{1}{l+1} \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i, \tag{5.13}$$

where $\eta(d, l) := \left(2^{\frac{d(l+1)-1}{2}} d^{-2}(l + 1)|D_K|^{\frac{1}{d}}\right)^{\frac{1}{l}}$.

Proof. Let $c \in \mathcal{O}_K$ and set $z_i = A$, for $1 \leq i \leq n$, where $A \geq 1$ is an integer parameter to be determined later. Then Theorem 5.4.4 implies that Algorithm 5.3.2 can be used to find all $x \in B_{c,M}$ such that

$$\sum_{i=1}^{n} \sigma_{w_i}^*(x, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i > \frac{dl}{2A}\left(\log M + \log \eta(d, l)\right) + \frac{A+1}{2(l+1)} \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i. \tag{5.14}$$

Setting $X = \frac{dl}{2}\left(\log M + \log \eta(d, l)\right)$ and $Y = \frac{1}{2(l+1)} \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i$, the right hand side of the inequality becomes $X/A + (A + 1)Y$.

The discriminant D_K is an integer. Thus

$$\eta(d,l) \geq \left(2^{d-\frac{1}{2}}d^{-2}\right)^{\frac{1}{l}} > \frac{1}{d}, \quad \text{for all } d \geq 1 \text{ and } l \geq 1.$$

Therefore, X is positive and the right hand side of (5.14) is minimised for $A = \sqrt{X/Y}$. By setting $A = \left\lceil \sqrt{X/Y} \right\rceil$, it follows that

$$\frac{X}{A} + (A+1)Y < \frac{X}{\sqrt{X/Y}} + (\sqrt{X/Y}+2)Y = 2\sqrt{XY} + 2Y.$$

Hence, with this choice of A, an element $x \in B_{c,M}$ satisfies (5.14) whenever

$$\sum_{i=1}^{n} \sigma^*_{w_i}(x,\mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i \geq 2\sqrt{XY} + 2Y.$$

Upon substitution of X and Y, this condition is found to be equivalent to (5.13). □

Remark 5.4.7. Corollary 5.4.6 requires that the parameter M, which determines the volume of $B_{c,M}$, satisfy the lower bound $M \geq d$. If $0 \leq M < d$, then $B_{c,M} = \{c\}$, for all $c \in \mathcal{O}_K$.

Corollary 5.4.6 admits bounds on the values of M for which Algorithm 5.3.2 can be used to find all elements in $B_{c,M}$ with norm containing a sufficiently large smooth factor. To end the section, two examples of these bounds are presented. In each example, the ideals $\mathfrak{p}_1, \dots, \mathfrak{p}_n$ are taken to consist of all prime ideals in $\mathcal{O}_K$ with norm less than 1000. Furthermore, the corresponding parameters $w_1, \dots, w_n$ are defined by

$$w_i = \left\lfloor \frac{\log 1000}{\log \mathfrak{N}\mathfrak{p}_i} \right\rfloor, \quad \text{for } 1 \leq i \leq n. \tag{5.15}$$

In each example, for $l = 49, 99$ and 999, an upper bound on M is provided such that, for any $c \in \mathcal{O}_K$, Algorithm 5.3.2 can be used to find all $x \in B_{c,M}$ with $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$. To begin, the simplest choice of number field K is considered, namely, $K = \mathbb{Q}$.

Example 5.4.8. Let $K = \mathbb{Q}$, then $d = 1$, $D_K = 1$ and $\mathcal{O}_K = \mathbb{Z}$. Moreover, $n = 168$ and $\prod_{i=1}^{168} \mathfrak{N}\mathfrak{p}_i^{w_i} \approx 2^{1437.9}$. Taking $l = 49$, for any $c \in \mathbb{Z}$ and $M \leq 2^{156.9}$, the algorithm performs reduction on a 50-dimensional lattice and returns all integers $x \in [c-M, c+M]$ such that $\prod_{i=1}^{168} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$. The maximum size of the searchable interval is approximately 955 times larger than that previously obtained by Boneh [25, Section 3]. However, the difference is due to improvements in the analysis and parameter selection, rather than algorithmic improvements. For $l = 99$ and $M \leq 2^{165.0}$, all integers $x \in [c-M, c+M]$ satisfying $\prod_{i=1}^{168} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$ are found. Similarly, all such integers are found for $l = 999$ and $M \leq 2^{172.5}$.

Example 5.4.9. Let $K = \mathbb{Q}(\alpha)$, where $\alpha \in \mathbb{C}$ is a root of the irreducible polynomial $x^3 - x^2 + 1$. Then $d = 3$, $D_K = -23$ and $\mathcal{O}_K = \mathbb{Z}[\alpha]$. Moreover, $n = 171$ and $\prod_{i=1}^{171} \mathfrak{N}\mathfrak{p}_i^{w_i} \approx 2^{1470.2}$. Taking $l = 49$, for any $c \in \mathcal{O}_K$ and $M \leq 2^{49.6}$, all $x \in B_{c,M}$ satisfying $\prod_{i=1}^{171} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$ are found. For $l = 49$, the algorithm performs reduction on a 150-dimensional lattice. For $l = 99$ and $M \leq 2^{52.3}$, all $x \in B_{c,M}$ satisfying $\prod_{i=1}^{171} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$ are found. Similarly, all such $x \in B_{c,M}$ are found for $l = 999$ and $M \leq 2^{54.8}$. The upper bounds on M for $l = 49, 99$ and 999 are approximately the cube root of the respective bounds in Example 5.4.8. This reduction is consistent with the increase in the product $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{w_i}$ and the observation that $|B_{c,M}|$ is approximately $\frac{2^{r_1}\pi^{r_2}}{\sqrt{|D_K|}} \frac{M^d}{d!} \approx 0.7M^3$, i.e, the cardinality of $B_{c,M}$ is approximately the cube of the number of integers in an interval of length $2M$.

5.4.2 Bounds on Smooth Algebraic Integers in Number Fields

In this section, combinatorially and algorithmically derived bounds on the existence of smooth algebraic integers in a number field are obtained. Explicit examples of the bounds are then provided. The results of this section are analogous to the combinatorial and algorithmic bounds on polynomial generation derived in Section 4.3 and Section 4.4.2, respectively. To begin, combinatorial arguments from Section 5.2 are extended to provide bounds on the number of algebraic integers in a number field with norm containing a large smooth factor.

Theorem 5.4.10. *Let K be a number field with $[K : \mathbb{Q}] = d$ that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$. Let $w_1, \ldots, w_n$ be positive integers and $z_1, \ldots, z_n$ be positive real numbers. Then for all $c \in \mathcal{O}_K$ and $M \geq d/2$, there exist at most l elements $x \in B_{c,M}$ such that*

$$\sum_{i=1}^{n} \sigma^*_{w_i}(x, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i \geq \sqrt{\left(\left(1 - \frac{1}{l}\right) d \log(2M/d) + \frac{1}{l} \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i\right) \sum_{i=1}^{n} z_i^2 w_i \log \mathfrak{N}\mathfrak{p}_i}. \tag{5.16}$$

Proof. Set $w = w_1 + \ldots + w_n$. Given $c \in \mathcal{O}_K$ and $M \geq d/2$, define $\mathcal{C}$ to be the set of all w-dimensional vectors

$$\left((x-c) + \mathfrak{p}_1, \ldots, (x-c) + \mathfrak{p}_1^{w_1}, \ldots, (x-c) + \mathfrak{p}_n, \ldots, (x-c) + \mathfrak{p}_n^{w_n}\right), \quad \text{for } x \in B_{c,M}.$$

Similarly, define

$$(t_1, \ldots, t_w) = (-c + \mathfrak{p}_1, \ldots, -c + \mathfrak{p}_1^{w_1}, \ldots, -c + \mathfrak{p}_n, \ldots, -c + \mathfrak{p}_n^{w_n}).$$

Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_w)$ be viewed as having n blocks, with the ith block equal to the w_i-dimensional vector $(\log \mathfrak{N}\mathfrak{p}_i, \ldots, \log \mathfrak{N}\mathfrak{p}_i)$, for $1 \leq i \leq n$. Similarly, let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_w)$ be viewed as having n blocks, with the ith block equal to the w_i-dimensional vector $(z_i \log \mathfrak{N}\mathfrak{p}_i, \ldots, z_i \log \mathfrak{N}\mathfrak{p}_i)$, for $1 \leq i \leq n$. Then Lemma 4.3.1 implies there exist at most l vectors $(x_1, \ldots, x_w) \in \mathcal{C}$ such that

$$\sum_{j: x_j = t_j} \beta_j \geq \sqrt{\left(\sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i - \left(1 - \frac{1}{l}\right) d(\mathcal{C})_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} z_i^2 w_i \log \mathfrak{N}\mathfrak{p}_i}, \tag{5.17}$$

where $d(\mathcal{C})_{\boldsymbol{\alpha}}$ is the minimum value, over all distinct pairs of vectors $(x_1, \ldots, x_w), (y_1, \ldots, y_w) \in \mathcal{C}$, of the sum $\sum_{j:x_i \neq y_j} \alpha_j$.

Suppose $x, y \in B_{c,M}$ are distinct. Then $(x-c)+\mathfrak{p}_i^j = (y-c)+\mathfrak{p}_i^j$, for indices $1 \leq i \leq n$ and $1 \leq j \leq w_i$, if and only if $x - y \in \mathfrak{p}_i^k$, for $1 \leq k \leq j$. Therefore, if $(x_1, \ldots, x_w)$ and $(y_1, \ldots, y_w)$ are the vectors in $\mathcal{C}$ that correspond to x and y respectively, then

$$\sum_{j:x_j \neq y_j} \alpha_j = \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i - \sum_{i=1}^{n} \sigma^*_{w_i}(x-y, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i \geq \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i - \log |N_K(x-y)|.$$

Furthermore,

$$|N_K(x-y)| \leq \frac{1}{d^d} \mathrm{size}(x-y)^d \leq \frac{1}{d^d} \left(\mathrm{size}(x-c) + \mathrm{size}(y-c)\right)^d \leq \left(\frac{2M}{d}\right)^d.$$

Since x and y were arbitrary distinct elements of $B_{c,M}$, it follows that $d(\mathcal{C})_{\boldsymbol{\alpha}} \geq \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i - d\log(2M/d)$. Similar arguments can be used to show that an element $x \in B_{c,M}$, and its corresponding vector $(x_1, \ldots, x_w) \in \mathcal{C}$, satisfy the relationship $\sum_{j:x_j = t_j} \beta_j = \sum_{i=1}^{n} \sigma^*_{w_i}(x, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i$. Therefore, if $x \in B_{c,M}$ satisfies (5.16), then its corresponding vector $(x_1, \ldots, x_w) \in \mathcal{C}$ satisfies (5.17). Hence, there exist at most l elements $x \in B_{c,M}$ such that (5.16) holds. □

Each output of Algorithm 5.3.2 occurs as a root over K of the polynomial $h \in \mathcal{O}_K[x]$ constructed in Step 3. Consequently, the number of outputs of the algorithm is bounded by the degree of h, which is at most l by construction. This observation is now used to obtain bounds on the number of elements in a ball $B_{c,M}$ with a large smooth factor, thus generalising those obtained by Boneh [25, Section 3.1] on smooth integers in short intervals. Analogous arguments were used in Section 4.4.2 to provide bounds on number field sieve polynomial generation. There the fact that every n-dimensional lattice $\Lambda \subset \mathbb{R}^n$ contains a nonzero vector $\boldsymbol{x}$ satisfying $\|\boldsymbol{x}\|_2 \leq \sqrt{\gamma_n} \det(\Lambda)^{1/n}$ was used to provide tighter bounds. By using similar arguments, the following bound is obtained:

Theorem 5.4.11. *Let K be a number field with $[K : \mathbb{Q}] = d$ that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$. Let $w_1, \ldots, w_n, z_1, \ldots, z_n, l$ be positive integers and M be a positive real number. For $c \in \mathcal{O}_K$, there exist at most l elements $x \in B_{c,M}$ such that*

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i) z_i} > \gamma_{d(l+1)}^{\frac{d}{2}} \, d^{-d} (l+1)^{\frac{d}{2}} \, M^{\frac{dl}{2}} \, \sqrt{|D_K|} \, \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{w_i \binom{z_i+1}{2}} \right)^{\frac{1}{l+1}}.$$

The proof of Theorem 5.4.11 is omitted since it only requires trivial changes to the proofs of Theorem 5.3.6 and Theorem 5.4.4. By careful selection of the weights $z_1, \ldots, z_n$, the following corollary is obtained:

Corollary 5.4.12. Let K be a number field with $[K : \mathbb{Q}] = d$ that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$. Let $w_1, \ldots, w_n$ be positive integers and $M \geq d$ be a real number. Then for any

Table 5.1: Bounds for Example 5.4.13

M	Corollary 5.4.12	Theorem 5.4.10
2^{100}	10	19
2^{110}	12	22
2^{120}	14	25
2^{130}	18	31
2^{140}	23	40
2^{150}	37	57
2^{160}	59	100
2^{170}	216	443

Table 5.2: Bounds for Example 5.4.14

M	Corollary 5.4.12	Theorem 5.4.10
2^{20}	6	13
2^{25}	7	15
2^{30}	9	17
2^{35}	12	21
2^{40}	16	27
2^{45}	23	37
2^{50}	41	61
2^{55}	171	193

$c \in \mathcal{O}_K$ and any integer $l \geq 1$, there exist at most l elements $x \in B_{c,M}$ such that

$$\sum_{i=1}^{n} \sigma^*_{w_i}(x, \mathfrak{p}_i) \log \mathfrak{N}\mathfrak{p}_i \geq \sqrt{\frac{dl}{l+1}\left(\log M + \log \eta(d,l)\right) \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i} + \frac{1}{l+1} \sum_{i=1}^{n} w_i \log \mathfrak{N}\mathfrak{p}_i, \tag{5.18}$$

where $\eta(d,l) := \left((2d)^{-2}\,(d(l+1)+4)\,(l+1)\,|D_K|^{\frac{1}{d}}\right)^{\frac{1}{l}}$.

The proof of the corollary is omitted as it is analogous to the proof of Corollary 5.4.6 of Theorem 5.4.4.

To end the section, examples of the bounds provided by Corollary 5.4.12 and Theorem 5.4.10 (with $z_1 = \ldots = z_n$) are compared. In each example, the ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ are taken to consist of all prime ideals in $\mathcal{O}_K$ with $\mathfrak{N}\mathfrak{p}_i \leq 1000$; and $w_1, \ldots, w_n$ are chosen according to (5.15). Then bounds on the number of elements $x \in B_{c,M}$ with $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$, are provided for various values of M.

Example 5.4.13. Let $K = \mathbb{Q}$. For various values of M, Table 5.1 contains bounds from Corollary 5.4.12 and Theorem 5.4.10 on the number of $x \in B_{c,M}$ such that $\prod_{i=1}^{168} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$, for any $c \in \mathbb{Z}$. For $M = 2^{99}$, Corollary 5.4.12 implies that any interval $I = [U, V]$ of length $V - U = 2^{100}$ contains at most 10 integers $x \in I \cap \mathbb{Z}$ such that $\prod_{i=1}^{168} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$. This improves upon the previous bound of 15 obtained by Boneh [25, Section 3.1]. Once again, the improvement here is only due to a more careful analysis. Note the large jump in the bounds for $M = 2^{160}$ and $M = 2^{170}$. For $M = 2^{171}, 2^{172}$ and 2^{173}, the bounds provided by Corollary 5.4.12 are 292, 450 and 976 respectively. This rapid growth is consistent with Example 5.4.8 where diminishing returns were observed as the lattice dimension increased.

Example 5.4.14. Let $K = \mathbb{Q}(\alpha)$, where $\alpha \in \mathbb{C}$ is a root of the irreducible polynomial $x^3 - x^2 + 1$. For various values of M, Table 5.2 contains bounds from Corollary 5.4.12 and Theorem 5.4.10 on the number of $x \in B_{c,M}$ such that $\prod_{i=1}^{171} \mathfrak{N}\mathfrak{p}_i^{\sigma^*_{w_i}(x,\mathfrak{p}_i)} \geq 2^{500}$, for any $c \in \mathcal{O}_K$.

Chapter 6

Conclusions and Future Research

The thesis was comprised of two parts: the first part, Chapters 2–4, concentrated on the polynomial selection problem; the second, Chapter 5, investigated smooth elements in number fields. In this concluding chapter, the original material presented in Chapters 3–5 of the thesis is summarised and potential avenues for future research discussed.

Chapter 3: Nonlinear Polynomial Selection

Properties of the orthogonal lattice were studied and used to develop precise criteria for the selection of geometric progressions in nonlinear algorithms. A family of geometric progressions containing those already used in existing algorithms was characterised. The characterisation was then used to extend existing nonlinear algorithms.

The partial characterisation of geometric progressions provided by Theorem 3.3.1 allowed extensions to be made to the length $d+1$ construction of Koo–Jo–Kwon. The extensions led to the development of a new polynomial generation algorithm (Algorithm 3.3.3) for which parameter selection was discussed. Using the algorithm, it was shown that pairs of degree $d \geq 2$ polynomials f_1 and f_2 may be found such that $\|f_i\|_{2,s} = O(N^{(1/d)(d^2-2d+2)/(d^2-d+2)})$, for $i = 1, 2$, where $s = \Theta(N^{(2/d)/(d^2-d+2)})$. The bound on the coefficients matches that obtained by Montgomery's algorithm, for $d = 2$ (which is optimal as a consequence of Corollary 2.1.4); the Prest–Zimmermann algorithm, for $d \geq 2$; and the Koo–Jo–Kwon algorithm, for $d \geq 2$. However, the increase in the parameter space of the new algorithm implies that a greater number of geometric progressions may be constructed for any given N. As a result, it is likely that polynomials with significantly smaller coefficients may be found with Algorithm 3.3.3 in practice.

Analogous to the extensions made to the length $d + 1$ construction, the length $d + 2$ construction of Koo, Jo and Kwon was also revisited and extended. This led to the development of a new polynomial generation algorithm (Algorithm 3.4.2) for which it was shown that pairs of degree $d \geq 3$ polynomials

f_1 and f_2 may be found such that $\|f_i\|_{2,s} = O(N^{(1/d)(d^2-4d+6)/(d^2-3d+6)})$, for $i = 1, 2$, where $s = \Theta(N^{(2/d)/(d^2-3d+6)})$. For $d = 3$, the exponent of N is optimal as a consequence of Corollary 2.1.4. However, as with the Koo–Jo–Kwon algorithms, the improvement in the exponent obtained over the length $d + 1$ algorithm is offset, in part, by the additional complexity of determining suitable parameters. The problem of determining parameters that meet the requirements of Section 3.4.1 requires further attention.

The area in which nonlinear algorithms have the greatest capacity for improvement is the construction of small geometric progressions. However, the algorithms in Chapter 3 both exploited features particular to the geometric progressions they used in order to guarantee that two degree d polynomials could be found. Therefore, the development of improved methods for generating geometric progressions may, in turn, necessitate the development of alternate methods to address this problem. In fact, the development of a mechanism that ensures two degree d polynomials can be found is of independent interest, since addressing the problem severely limited the choice of skews in analyses of Section 3.3.1 and Section 3.4.1. Finally, such a mechanism may also aid in the development of nonlinear algorithms that produce polynomial pairs with distinct degrees.

Chapter 4: An Approach to Polynomial Selection

A new approach to the problem of generating polynomials with a good combination of size and root properties was developed. The approach, which shares more in common with the list decoding algorithms for algebraic codes than previous polynomial generation algorithms, exploits much of the underlying algebraic structure of the polynomial selection problem. An initial realisation of the approach (Algorithm 4.4.2) was provided and analysed. In addition, combinatorially and algorithmically derived bounds on the existence of number field sieve polynomials with small coefficients and good non-projective root properties were obtained. Finally, possible improvements and generalisations of the new approach were discussed, demonstrating its flexibility.

The development of the approach to polynomial generation was underpinned by the study of the divisibility properties of univariate resultants. By furthering this study, it may possible to exploit more of the underlying algebraic structure of the problem. In particular, it may be possible to modify the approach to account for projective root properties. In addition to potentially affording improvements to the approach to polynomial generation, the study of the divisibility properties of univariate resultants is of independent interest. It is likely that the results of Section 4.2 are open to generalisation.

The combinatorial results of Section 4.3 provide the first examples of bounds on the existence of number field sieve polynomials with constrained size and root properties. In Section 4.4.1, the bounds were useful in assessing the performance of Algorithm 4.4.2. However, bounds on the existence of number field polynomials are of independent interest. For example, such bounds may assist in determining at which point to terminate the search for polynomials and progress to the sieve stage of the number field sieve, thus helping to minimise wasted effort. Explicit examples of the bounds derived in Section 4.3 were provided in Example 4.3.5. There the bounds were only applicable with extremely restrictive requirements on the size of the polynomials. It may be possible to extend these bound to greater ranges of interest by applying finer combinatorial arguments (see Remark 4.3.3). Corollary 4.3.6 provided a condition under which the approach to polynomial generation described in Section 4.1 is combinatorially feasible. The natural question on the optimality of the condition arises immediately. The problem of answering this question is left for future investigation.

The realisation of the approach to polynomial generation yielded an algorithm (Algorithm 4.4.2) that differs significantly from previous algorithms for polynomial generation. The most significant difference, and a major feature of the approach, is the algorithm's simultaneous, rather than sequential, consideration of size and root properties. Additionally, for each choice of parameters, the algorithm is guaranteed to find *all* polynomials with sufficiently good size and root properties. In Section 4.4.1, parameter selection for Algorithm 4.4.2 in the presence of real weights was considered. Comparison with the combinatorial bounds suggested that the algorithm does not perform optimally. Moreover, examples of parameter selection, under a natural choice of weights, suggested that the algorithms complexity is too large to justify its practical application. To address this problem, possible avenues for generalising the approach of Section 4.1, and improving its realisation, were described in Section 4.5. There it was shown that a small departure from the approach (for example, the introduction of a special-$\mathfrak{q}$) can have a large effect on its complexity.

In practical circumstances, the average-case behaviour of a polynomial generation algorithm is relevant. Thus, it may be worthwhile to investigate average-case behaviour of Algorithm 4.4.2, either theoretically or by computational experiments. The worst-case behaviour of Algorithm 4.4.2 was estimated in Theorem 4.4.3, which may be rather pessimistic when compared with the average-case behaviour. For example, the proof of Theorem 4.4.3 applies a bound on the first basis vector in an LLL-reduced basis (Theorem 3.1.2), which holds for all lattices. However, Algorithm 4.4.2 applies LLL-reduction to lattices which are not random and, in some sense, highly structured. Thus, the general bound on the shortest vector may not reflect average-case behaviour. A second example is provided by the bound on the resultant from Lemma 2.1.3, which is also applied in the proof of Theorem 4.4.3. At is core, Lemma 2.1.3 is simply an application of Hadamard's inequality. Although Hadamard's inequality is tight, in the sense that it is attainable for all dimensions, the inequality does not reflect average-case behaviour [1, 2]: roughly speaking and for large n, the determinant of an $n \times n$ real matrix is smaller than the Hadamard bound by roughly a factor of $e^{-n/2}\sqrt[4]{4e}$ on average [2, Lemma 3.2]. Thus, it may hold that the upper bound on the resultant in Lemma 2.1.3 is rather pessimistic on average, for polynomials with large degree sum. Further investigation is required to determine whether this is true, since the proof of Lemma 2.1.3 applies Hadamard's inequality to bound the determinant Sylvester matrices only.

Very little is know about the existence of number field sieve polynomials with specified size and roots properties. Consequently, for many of the algorithms discussed in Chapter 4, it is not clear under which choices of parameters an algorithm will return a non-empty set of polynomials. This problem may have negative consequences in practical circumstances (see Remark 4.4.5). As a result, it is of theoretical and practical interest to provide existence results for number field sieve polynomials with specified size and roots properties.

An algorithm based on the approach of Section 4.1 is, in principle, capable of generating more than two number field sieve polynomials with a common root modulo N. Therefore, the approach may naturally lend itself to addressing the polynomial selection problem for multiple polynomial versions of the number field sieve [40, 55], for which little is known. The utility of the approach in this setting requires further investigation.

The methods of Section 4.5.3 were noted to have potential applications outside the polynomial selection problem. In particular, the possibility of applying the methods in new attacks on RSA, and to solve factorisation problems, was raised. Apart from being of independent interest, furthering the development of methods from Section 4.5.3 within either of these settings may be beneficial to the future development of polynomial generation algorithms.

Chapter 5: Smooth Elements in Number Fields

The list decoding for CRT codes was generalised to number fields, providing the first algorithm (Algorithm 5.3.2) for solving the weighted list decoding problem for NF-codes. The decoding algorithm then played a central role in the development of an algorithm for finding algebraic integers in a number field with norm containing a large smooth factor. Finally, bounds on the existence of such elements were derived.

For number fields other than $\mathbb{Q}$, the error-correction performance of the list decoding algorithm was shown not to meet the bound for which decoding is combinatorially feasible. The difference between the error-correction performance and the combinatorial bound results from the loss of structure that occurs when the space of polynomials of bounded degree in $\bigcap_{i=1}^{n} I_i^{z_i}$ is treated as a $\mathbb{Z}$-module rather than an $\mathcal{O}_K$-module. This observation explains why the difference depends on the degree of the number field only. It may be possible to obtain an algorithm that meets the combinatorial bound by considering this additional structure. Unfortunately, there is no known analogue of the LLL algorithm for arbitrary $\mathcal{O}_K$-modules that admits bounds on the lengths of the basis vectors (however, some results have been obtained in this direction [58, 127, 60]). Hence, exploiting the full structure of $\mathcal{O}_K$-modules in this setting appears to be a nontrivial problem.

In Chapter 5, smooth elements in number fields were studied in a general setting, i.e., not within the context of the number field sieve. A partial connection with the sieve stage of the number field sieve is established by restricting attention to first degree prime ideals. However, the results of the chapter are stated for elements of a ball $B_{c,M}$, rather than elements corresponding to the sieve region. Therefore, the performance of the algorithm for finding smooth elements and sieving are not directly comparable. As a result, further investigation is required to determine whether it is worthwhile to use an approach based on the methods of Chapter 5 in place of sieving. In particular, it may be worthwhile to investigate average-case behaviour of the approach, which is relevant in practical circumstances, and to determine for which parameters the algorithms discussed in Section 5.4.1 have nonempty output.

A major theme throughout the thesis was the utility of applying decoding algorithms for algebraic error-correcting codes to problems of finding elements in a ring with a smooth representation. In addition to the cited examples of Cheng and Wan [35], and Boneh [25], the results of Chapter 5 and Chapter 4 provided further evidence toward this claim.

References

[1] John Abbott, Manuel Bronstein, and Thom Mulders. Fast deterministic computation of determinants of dense matrices. In *Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (Vancouver, BC)*, pages 197–204 (electronic), New York, 1999. ACM.

[2] John Abbott and Thom Mulders. How tight is Hadamard's bound? *Experiment. Math.*, 10(3):331–336, 2001.

[3] Leonard M. Adleman. Factoring numbers using singular integers. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 64–71, New York, NY, USA, 1991. ACM.

[4] A.C. Aitken. *Determinants and Matrices.* Oliver and Boyd, Edinburgh, ninth edition, 1956.

[5] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 601–610 (electronic), New York, 2001. ACM.

[6] W.R. Alford and Carl Pomerance. Implementing the self initializing quadratic sieve on a distributed network. In *Number Theoretic and Algebraic Methods in Computer Science, Proc. of Int'l Moscow Conference, June-July, 1993, (A. J. van der Poorten, I. Shparlinski, H. G. Zimmer, eds.)*, pages 163–174. World Scientific, 1996.

[7] Kazumaro Aoki, Jens Franke, Thorsten Kleinjung, Arjen K. Lenstra, and Dag Arne Osvik. A kilobit special number field sieve factorization. In *Advances in cryptology—ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, Berlin, 2007.

[8] François Apéry and Jean-Pierre Jouanolou. *Élimination – Le cas d'une variable.* Mathématiques LMD–Master. Hermann, Paris, 2006.

[9] Tom M. Apostol. *Introduction to analytic number theory.* Springer-Verlag, New York, 1976. Undergraduate Texts in Mathematics.

[10] Eric Bach and René Peralta. Asymptotic semismoothness probabilities. *Math. Comp.*, 65(216):1701–1715, 1996.

[11] Friedrich Bahr, M. Böhm, Jens Franke, and Thorsten Kleinjung. Factorization of RSA-200. http://www.loria.fr/~zimmerma/records/rsa200, May 2005.

[12] Shi Bai, Richard P. Brent, and Emmanuel Thomé. Root optimization of polynomials in the number field sieve. ArXiv e-Print archive, `arXiv:1212.1958 [math.NT]`, December 2012. http://arxiv.org/abs/1212.1958.

[13] Aurélie Bauer and Antoine Joux. Toward a rigorous variation of Coppersmith's algorithm on three variables. In *Advances in cryptology—EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Comput. Sci.*, pages 361–378. Springer, Berlin, 2007.

[14] Karim Belabas. A relative van Hoeij algorithm over number fields. *J. Symbolic Comput.*, 37(5):641–668, 2004.

[15] Karim Belabas. Topics in computational algebraic number theory. *J. Théor. Nombres Bordeaux*, 16(1):19–63, 2004.

[16] Karim Belabas, Mark van Hoeij, Jürgen Klüners, and Allan Steel. Factoring polynomials over global fields. *J. Théor. Nombres Bordeaux*, 21(1):15–39, 2009.

[17] Edward A. Bender and E. Rodney Canfield. An approximate probabilistic model for structured Gaussian elimination. *J. Algorithms*, 31(2):271–290, 1999.

[18] Daniel J. Bernstein. Reducing lattice bases to find small-height values of univariate polynomials. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 421–446. Cambridge Univ. Press, Cambridge, 2008.

[19] Daniel J. Bernstein and Arjen K. Lenstra. A general number field sieve implementation. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 103–126. Springer, Berlin, 1993.

[20] Étienne Bézout. Recherches sur le degré des équations résultantes de l'évanouissement des inconnues, et sur les moyens qu'il convient d'employer pour trouver ces équations. *Mém. Acad. Roy. Sci. Paris*, pages 288–338, 1764.

[21] Johannes Blömer and Alexander May. Low secret exponent RSA revisited. In *Cryptography and lattices (Providence, RI, 2001)*, volume 2146 of *Lecture Notes in Comput. Sci.*, pages 4–19. Springer, Berlin, 2001.

[22] Henk Boender. The number of relations in the quadratic sieve algorithm. Technical report, Department of Numerical Mathematics, Centrum voor Wiskunde en Informatica, Amsterdam, 1996.

[23] Henk Boender. *Factoring large integers with the quadratic sieve.* PhD thesis, University of Leiden, 1997.

[24] Henk Boender and Herman J. J. te Riele. Factoring integers with large-prime variations of the quadratic sieve. *Experiment. Math.*, 5(4):257–273, 1996.

[25] Dan Boneh. Finding smooth integers in short intervals using CRT decoding. *J. Comput. System Sci.*, 64(4):768–784, 2002. Special issue on STOC 2000 (Portland, OR).

[26] John Brillhart, Michael Filaseta, and Andrew Odlyzko. On an irreducibility theorem of A. Cohn. *Canad. J. Math.*, 33(5):1055–1059, 1981.

[27] J. A. Buchmann and Hendrik W. Lenstra, Jr. Approximating rings of integers in number fields. *J. Théor. Nombres Bordeaux*, 6(2):221–260, 1994.

[28] Johannes Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. In *Séminaire de Théorie des Nombres, Paris 1988–1989*, volume 91 of *Progr. Math.*, pages 27–41. Birkhäuser Boston, Boston, MA, 1990.

[29] J. P. Buhler, Hendrik W. Lenstra, Jr., and Carl Pomerance. Factoring integers with the number field sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 50–94. Springer, Berlin, 1993.

[30] E. R. Canfield, Paul Erdős, and Carl Pomerance. On a problem of Oppenheim concerning "factorisatio numerorum". *J. Number Theory*, 17(1):1–28, 1983.

[31] J. W. S. Cassels. *An introduction to the geometry of numbers.* Springer-Verlag, Berlin, 1971. Second printing, corrected, Die Grundlehren der mathematischen Wissenschaften, Band 99.

[32] 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. In *Proceedings of the 19th international conference on Theory and application of cryptographic techniques*, EUROCRYPT'00, pages 1–18, 2000.

[33] Stefania Cavallar, Walter Lioen, Herman te Riele, Bruce Dodson, Arjen Lenstra, Paul Leyland, Peter L. Montgomery, Brian Murphy, and Paul Zimmermann. Factorization of RSA-140 using the number field sieve. In *In Advances in Cryptology, Asiacrypt'99*, pages 195–207. Springer-Verlag, 1999.

[34] A. Cayley. Note sur la méthode d'élimination de bezout. *J. Reine Angew. Math.*, 53:366–367, 1857.

[35] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of Reed-Solomon codes. *SIAM J. Comput.*, 37(1):195–209 (electronic), 2007.

[36] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics.* Springer-Verlag, Berlin, 1993.

[37] Henri Cohen. *Advanced topics in computational number theory*, volume 193 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2000.

[38] Henry Cohn and Nadia Heninger. Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. ArXiv e-Print archive, `arXiv:1008.1284v1 [math.NT]`, August 2010. http://arxiv.org/abs/1008.1284.

[39] Scott P. Contini. Factoring integers with the self-initializing quadratic sieve. Master's thesis, U. Georgia, 1997.

[40] Don Coppersmith. Modifications to the number field sieve. *J. Cryptology*, 6(3):169–180, 1993.

[41] Don Coppersmith. Solving linear equations over GF(2): block Lanczos algorithm. *Linear Algebra Appl.*, 192:33–60, 1993. Computational linear algebra in algebraic and related problems (Essen, 1992).

[42] Don Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Math. Comp.*, 62(205):333–350, 1994.

[43] Don Coppersmith. Finding a small root of a univariate modular equation. In *Advances in cryptology—EUROCRYPT '96 (Saragossa, 1996)*, volume 1070 of *Lecture Notes in Comput. Sci.*, pages 155–165. Springer, Berlin, 1996.

[44] Jean-Marc Couveignes. Computing a square root for the number field sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 95–102. Springer, Berlin, 1993.

[45] 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 *Advances in cryptology—ASIACRYPT '96 (Kyongju)*, volume 1163 of *Lecture Notes in Comput. Sci.*, pages 382–394. Springer, Berlin, 1996.

[46] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms.* Undergraduate Texts in Mathematics. Springer, New York, third edition, 2007. An introduction to computational algebraic geometry and commutative algebra.

[47] Nicholas Coxon. On nonlinear polynomial selection for the number field sieve. ArXiv e-Print archive, `arXiv:1109.6398 [math.NT]`, September 2010. http://arxiv.org/abs/1109.6398.

[48] Nicholas Coxon. List decoding of number field codes. *Designs, codes and cryptography*, 2013. doi:10.1007/s10623-013-9803-x.

[49] Richard Crandall and Carl Pomerance. *Prime numbers: A computational perspective.* Springer, New York, second edition, 2005.

[50] Alicia Dickenstein and Ioannis Z. Emiris, editors. *Solving polynomial equations*, volume 14 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2005. Foundations, algorithms, and applications.

[51] K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Ark. Mat., Astronomi och Fysik*, 22A(10):1–4, 1930.

[52] John D. Dixon. Asymptotically fast factorization of integers. *Math. Comp.*, 36(153):255–260, 1981.

[53] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[54] Marije Elkenbracht-Huizing. An implementation of the number field sieve. *Experiment. Math.*, 5(3):231–253, 1996.

[55] Marije Elkenbracht-Huizing. A multiple polynomial general number field sieve. In *Algorithmic number theory (Talence, 1996)*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 99–114. Springer, Berlin, 1996.

[56] R.-M. Elkenbracht-Huizing, Peter L. Montgomery, R. D. Silverman, R. K. Wackerbarth, and S. S. Wagstaff, Jr. The number field sieve on many computers. In *Number theory (Ottawa, ON, 1996)*, volume 19 of *CRM Proc. Lecture Notes*, pages 81–85. Amer. Math. Soc., Providence, RI, 1999.

[57] Leonhard Euler. *Introductio in analysin infinitorum, Tom. 2.* Lausanne, 1748.

[58] C. Fieker and M. E. Pohst. On lattices over number fields. In *Algorithmic number theory (Talence, 1996)*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 133–139. Springer, Berlin, 1996.

[59] Claus Fieker and Carsten Friedrichs. On reconstruction of algebraic numbers. In *Algorithmic number theory (Leiden, 2000)*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 285–296. Springer, Berlin, 2000.

[60] Ying Hung Gan, Cong Ling, and Wai Ho Mow. Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection. *IEEE Trans. Signal Process.*, 57(7):2701–2710, 2009.

[61] I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Mathematics: Theory & Applications. Birkhäuser Boston Inc., Boston, MA, 1994.

[62] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. *IEEE Trans. Inform. Theory*, 46(4):1330–1338, 2000.

[63] Domingo Gomez, Jaime Gutierrez, Álvar Ibeas, and David Sevilla. Common factors of resultants modulo p. *Bull. Aust. Math. Soc.*, 79(2):299–302, 2009.

[64] Daniel M. Gordon. Discrete logarithms in GF(p) using the number field sieve. *SIAM J. Discrete Math.*, 6(1):124–138, 1993.

[65] Jason E. Gower. Rotations and translations of number field sieve polynomials. In *Advances in Cryptology - ASIACRYPT 2003*, pages 302–310, 2003.

[66] Andrew Granville. Smooth numbers: computational number theory and beyond. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 267–323. Cambridge Univ. Press, Cambridge, 2008.

[67] Venkatesan Guruswami. Constructions of codes from number fields. *IEEE Trans. Inform. Theory*, 49(3):594–603, 2003.

[68] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes: Winning Thesis of the 2002 ACM Doctoral Dissertation Competition*, volume 3282 of *Lecture Notes in Computer Science*. Springer, New York, 2004.

[69] Venkatesan Guruswami, Amit Sahai, and Madhu Sudan. "Soft-decision" decoding of Chinese remainder codes. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 159–168. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

[70] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999.

[71] Venkatesan Guruswami and Madhu Sudan. Extensions to the Johnson bound (manuscript). 2001.

[72] Jacques Hadamard. Résolution dune question relative aux déterminants. *Bull. des Sci. Math.*, 17:240–246, 1893.

[73] Helmut Hasse. Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik. *J. Reine Angew. Math.*, 175:50–54, 1936.

[74] George Havas, Bohdan S. Majewski, and Keith R. Matthews. Extended GCD and Hermite normal form algorithms via lattice basis reduction. *Experiment. Math.*, 7(2):125–136, 1998.

[75] Mathias Herrmann, Alexander May, and Maike Ritzenhofen. Polynomial selection using lattices. Slides presented at the CITS Workshop on Factoring Large Integers, Ruhr-Universität Bochum, Germany, 2009. 65 pages, available at http://www.cits.rub.de/imperia/md/content/may/factoring2009/herrmann.pdf

[76] M. J. Hinek. New partial key exposure attacks on RSA revisited. Technical report, Centre for Applied Cryptographic Research, University of Waterloo, March 2004.

[77] M. J. Hinek. Small private exponent partial key-exposure attacks on multiprime RSA. Technical report, Centre for Applied Cryptographic Research, University of Waterloo, May 2005.

[78] Nicholas Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and coding (Cirencester, 1997)*, volume 1355 of *Lecture Notes in Computer Science*, pages 131–142. Springer, Berlin, 1997.

[79] Nicholas Howgrave-Graham. *Computational Mathematics Inspired by RSA*. PhD thesis, University of Bath (UK), 1998.

[80] C.G.J. Jacobi. De eliminatione variabilis e duabus aequationibus algebraicis. *J. Reine Angew. Math.*, 15:101–124, 1836.

[81] Přemysl Jedlička. Integral minimisation improvement for Murphy's polynomial selection algorithm. *An. Ştiinţ. Univ. "Ovidius" Constanţa Ser. Mat.*, 18(2):125–130, 2010.

[82] Ellen Jochemsz and Alexander May. A polynomial time attack on RSA with private CRT-exponents smaller than $N^{0.073}$. In *Advances in cryptology—CRYPTO 2007*, volume 4622 of *Lecture Notes in Comput. Sci.*, pages 395–411. Springer, Berlin, 2007.

[83] Selmer M. Johnson. A new upper bound for error-correcting codes. *IRE Trans.*, IT-8:203–207, 1962.

[84] Selmer M. Johnson. Improved asymptotic bounds for error-correcting codes. *IEEE Trans. Information Theory*, IT-9:198–205, 1963.

[85] Jean-Pierre Jouanolou. Le formalisme du résultant. *Adv. Math.*, 90(2):117–263, 1991.

[86] Antoine Joux, Reynald Lercier, Nigel Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in cryptology—CRYPTO 2006*, volume 4117 of *Lecture Notes in Comput. Sci.*, pages 326–344. Springer, Berlin, 2006.

[87] Erich Kaltofen. On the complexity of finding short vectors in integer lattices. In *Computer algebra (London, 1983)*, volume 162 of *Lecture Notes in Comput. Sci.*, pages 236–244. Springer, Berlin, 1983.

[88] Ravi Kannan. Algorithmic geometry of numbers. In *Annual review of computer science, Vol. 2*, pages 231–267. Annual Reviews, Palo Alto, CA, 1987.

[89] Deepak Kapur and Tushar Saxena. Comparison of various multivariate resultant formulations. In *Proceedings of the 1995 international symposium on symbolic and algebraic computation*, ISSAC '95, pages 187–194, New York, NY, USA, 1995. ACM.

[90] Thorsten Kleinjung. Polynomial selection. Slides presented at the CADO workshop, Nancy, France, 2008. 30 pages, available at http://cado.gforge.inria.fr/workshop/slides/kleinjung.pdf.

[91] Thorsten Kleinjung. On polynomial selection for the general number field sieve. *Math. Comp.*, 75(256):2037–2047 (electronic), 2006.

[92] Thorsten Kleinjung, Kazumaro Aoki, Jens Franke, Arjen Lenstra, Emmanuel Thomé, Joppe Bos, Pierrick Gaudry, Alexander Kruppa, Peter Montgomery, Dag Arne Osvik, Herman te Riele, Andrey Timofeev, and Paul Zimmermann. Factorization of a 768-bit RSA modulus. In *CRYPTO 2010 Advances in Cryptology - CRYPTO 2010 (Santa Barbara, USA, 2010), T. Rabin, Ed.*, volume 6223 of *Lecture Notes in Comput. Sci.*, pages 333–350. Springer-Verlag, Berlin, 2010.

[93] Donald E. Knuth. *The art of computer programming. Vol. 2: Seminumerical algorithms.* Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont, 1969.

[94] Donald E. Knuth. *The art of computer programming. Vol. 2.* Addison-Wesley Publishing Co., Reading, Mass., second edition, 1981. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

[95] Donald E. Knuth and Luis Trabb Pardo. Analysis of a simple factorization algorithm. *Theoret. Comput. Sci.*, 3(3):321–348, 1976/77.

[96] Sergei V. Konyagin and Igor E. Shparlinski. *Character sums with exponential functions and their applications*, volume 136 of *Cambridge Tracts in Mathematics.* Cambridge University Press, Cambridge, 1999.

[97] Namhun Koo, Gooc Hwa Jo, and Soonhak Kwon. On nonlinear polynomial selection and geometric progression (mod N) for number field sieve. Cryptology ePrint Archive, Report 2011/292, 2011. http://eprint.iacr.org/2011/292.pdf.

[98] B. LaMacchia and A. Odlyzko. Solving large sparse linear systems over finite fields. In Alfred Menezes and Scott Vanstone, editors, *Advances in Cryptology-CRYPT0 90*, volume 537 of *Lecture Notes in Computer Science*, pages 109–133. Springer Berlin / Heidelberg, 1991.

[99] Serge Lang. *Algebra*, volume 211 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, third edition, 2002.

[100] D. H. Lehmer and R. E. Powers. On factoring large numbers. *Bull. Amer. Math. Soc.*, 37(10):770–776, 1931.

[101] Gottfried Wilhelm Leibniz. Draft letter to Tschirnhaus (1683). In *Der Briefwechsel von Gottfried Wilhelm Leibniz mit Mathematikern*, Herausgegeben von C. I. Gerhardt, pages xxviii+761 pp. (1 insert). Georg Olms Verlagsbuchhandlung, Hildesheim, 1962.

[102] Arjen K. Lenstra and Hendrik. W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1993.

[103] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[104] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 11–42. Springer, Berlin, 1993.

[105] Arjen K. Lenstra and M. S. Manasse. Factoring with two large primes. *Math. Comp.*, 63(208):785–798, 1994.

[106] Hendrik W. Lenstra, Jr. Codes from algebraic number fields. In *Mathematics and computer science, II (Amsterdam, 1986)*, volume 4 of *CWI Monogr.*, pages 95–104. North-Holland, Amsterdam, 1986.

[107] Hendrik W. Lenstra, Jr. Algorithms in algebraic number theory. *Bull. Amer. Math. Soc. (N.S.)*, 26(2):211–244, 1992.

[108] Hendrik W. Lenstra, Jr. Lattices. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 127–181. Cambridge Univ. Press, Cambridge, 2008.

[109] Hendrik W. Lenstra, Jr. Solving the Pell equation. In *Algorithmic number theory: lattices, number fields, curves and cryptography*, volume 44 of *Math. Sci. Res. Inst. Publ.*, pages 1–23. Cambridge Univ. Press, Cambridge, 2008.

[110] Hendrik W. Lenstra, Jr. and Carl Pomerance. A rigorous time bound for factoring integers. *J. Amer. Math. Soc.*, 5(3):483–516, 1992.

[111] Paul Leyland, Arjen Lenstra, Bruce Dodson, Alec Muffett, and Sam Wagstaff. MPQS with three large primes. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 446–460. Springer, Berlin, 2002.

[112] Francis S. Macaulay. *The algebraic theory of modular systems*, volume 19 of *Cambridge tracts in mathematics and mathematical physics*. Cambridge University Press, Cambridge, 1916.

[113] David M. Mandelbaum. On a class of arithmetic codes and a decoding algorithm. *IEEE Trans. Information Theory*, IT-22(1):85–88, 1976.

[114] Daniel A. Marcus. *Number fields*. Springer-Verlag, New York, 1977. Universitext.

[115] Jacques Martinet. *Perfect lattices in Euclidean spaces*, volume 327 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2003.

[116] Alexander May. *New RSA Vulnerabilities Using Lattice Reduction Methods.* PhD thesis, University of Paderborn, 2003.

[117] Alexander May. Using LLL-reduction for solving RSA and factorization problems. In Phong Q. Nguyen and Brigitte Vallée, editors, *The LLL Algorithm*, Information Security and Cryptography, pages 315–348. Springer Berlin Heidelberg, 2010.

[118] John Milnor and Dale Husemoller. *Symmetric bilinear forms.* Springer-Verlag, New York, 1973. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 73.

[119] Peter L. Montgomery. Small geometric progressions modulo n. Unpublished note of 2 pages, December 1993.

[120] Peter L. Montgomery. Square roots of products of algebraic numbers. In *Mathematics of Computation 1943–1993: a half-century of computational mathematics (Vancouver, BC, 1993)*, volume 48 of *Proc. Sympos. Appl. Math.*, pages 567–571. Amer. Math. Soc., Providence, RI, 1994.

[121] Peter L. Montgomery. A block Lanczos algorithm for finding dependencies over GF(2). In *Advances in cryptology—EUROCRYPT '95 (Saint-Malo, 1995)*, volume 921 of *Lecture Notes in Comput. Sci.*, pages 106–120. Springer, Berlin, 1995.

[122] Peter L. Montgomery. Searching for higher-degree polynomials for the general number field sieve. Power-Point presentation, 34 pages, available at http://www.ipam.ucla.edu/publications/scws1/scws1_6223.ppt, 2006.

[123] Michael A. Morrison and John Brillhart. A method of factoring and the factorization of F_7. *Math. Comp.*, 29:183–205, 1975. Collection of articles dedicated to Derrick Henry Lehmer on the occasion of his seventieth birthday.

[124] Brian Murphy and Richard P. Brent. On quadratic polynomials for the number field sieve. In *Computing theory '98 (Perth)*, volume 20 of *Aust. Comput. Sci. Commun.*, pages 199–213. Springer, Singapore, 1998.

[125] Brian A. Murphy. Modelling the yield of number field sieve polynomials. In *Algorithmic number theory (Portland, OR, 1998)*, volume 1423 of *Lecture Notes in Comput. Sci.*, pages 137–150. Springer, Berlin, 1998.

[126] Brian A. Murphy. *Polynomial selection for the number field sieve integer factorisation algorithm.* PhD thesis, Australian National University, July 1999.

[127] Huguette Napias. A generalization of the LLL-algorithm over Euclidean rings or orders. *J. Théor. Nombres Bordeaux*, 8(2):387–396, 1996.

[128] Władysław Narkiewicz. *Elementary and analytic theory of algebraic numbers.* Springer Monographs in Mathematics. Springer-Verlag, Berlin, third edition, 2004.

[129] Phong Nguyen. A Montgomery-like square root for the number field sieve. In *Algorithmic number theory (Portland, OR, 1998)*, volume 1423 of *Lecture Notes in Comput. Sci.*, pages 151–168. Springer, Berlin, 1998.

[130] Phong Nguyen and Jacques Stern. Merkle-Hellman revisited: a cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Advances in cryptology—CRYPTO '97 (Santa Barbara, CA, 1997)*, volume 1294 of *Lecture Notes in Comput. Sci.*, pages 198–212. Springer, Berlin, 1997.

[131] Phong Q. Nguyen and Damien Stehlé. Floating-point LLL revisited. In *Advances in cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Comput. Sci.*, pages 215–233. Springer, Berlin, 2005.

[132] Phong Q. Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3):874–903, 2009.

[133] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Trans. Algorithms*, 5(4):Art. 46, 48, 2009.

[134] Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL algorithm: Survey and Applications.* Information Security and Cryptography. Springer-Verlag, Berlin, 2010.

[135] J. E. Nymann. The distribution of relatively r-prime integers in residue classes. *Rocky Mountain J. Math.*, 22(4):1473–1482, 1992.

[136] A. M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In *Advances in cryptology (Paris, 1984)*, volume 209 of *Lecture Notes in Comput. Sci.*, pages 224–314. Springer, Berlin, 1985.

[137] René Peralta. A quadratic sieve on the n-dimensional cube. In *Advances in cryptology—CRYPTO '92 (Santa Barbara, CA, 1992)*, volume 740 of *Lecture Notes in Comput. Sci.*, pages 324–332. Springer, Berlin, 1993.

[138] Michael Peterson and Chris Monico. $\mathbb{F}_2$ Lanczos revisited. *Linear Algebra Appl.*, 428(4):1135–1150, 2008.

[139] M. Pohst and H. Zassenhaus. *Algorithmic algebraic number theory*, volume 30 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 1989.

[140] Michael E. Pohst. A modification of the LLL reduction algorithm. *J. Symbolic Comput.*, 4(1):123–127, 1987.

[141] J. M. Pollard. Factoring with cubic integers. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 4–10. Springer, Berlin, 1993.

[142] J. M. Pollard. The lattice sieve. In *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.*, pages 43–49. Springer, Berlin, 1993.

[143] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In *Computational methods in number theory, Part I*, volume 154 of *Math. Centre Tracts*, pages 89–139. Math. Centrum, Amsterdam, 1982.

[144] Carl Pomerance. The quadratic sieve factoring algorithm. In *Advances in cryptology (Paris, 1984)*, volume 209 of *Lecture Notes in Comput. Sci.*, pages 169–182. Springer, Berlin, 1985.

[145] Carl Pomerance. The number field sieve. In *Mathematics of Computation 1943–1993: a half-century of computational mathematics (Vancouver, BC, 1993)*, volume 48 of *Proc. Sympos. Appl. Math.*, pages 465–480. Amer. Math. Soc., Providence, RI, 1994.

[146] Carl Pomerance. The role of smooth numbers in number-theoretic algorithms. In *Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Zürich, 1994)*, pages 411–422, Basel, 1995. Birkhäuser.

[147] Carl Pomerance and J. W. Smith. Reduction of huge, sparse matrices over finite fields via created catastrophes. *Experiment. Math.*, 1(2):89–94, 1992.

[148] Carl Pomerance, J. W. Smith, and Randy Tuler. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. *SIAM J. Comput.*, 17(2):387–403, 1988. Special issue on cryptography.

[149] Thomas Prest and Paul Zimmermann. Non-linear polynomial selection for the number field sieve. *J. Symb. Comput.*, 47(4):401–409, April 2012.

[150] Xavier-François Roblot. Polynomial factorization algorithms over number fields. *J. Symbolic Comput.*, 38(5):1429–1443, 2004.

[151] Oliver Schirokauer. The number field sieve for integers of low weight. *Math. Comp.*, 79(269):583–602, 2010.

[152] Katja Schmidt-Samoa. Das number field sieve: Entwicklung, varianten und erfolge. Diploma thesis, Universität Kaiserslautern, March 2002.

[153] Alexander Schrijver. *Theory of linear and integer programming.* Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, 1986. A Wiley-Interscience Publication.

[154] Igor E. Shparlinski and Ron Steinfeld. Chinese remaindering for algebraic numbers in a hidden field. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 349–356. Springer, Berlin, 2002.

[155] V. Sidorenko, G. Schmidt, E. Gabidulin, M. Bossert, and V. Afanassiev. On polyalphabetic block codes. In *Proc. IEEE ISOC ITW2005 on Coding and Complexity*, pages 207–210, 2005.

[156] Robert D. Silverman. The multiple polynomial quadratic sieve. *Math. Comp.*, 48(177):329–339, 1987.

[157] Robert D. Silverman. Optimal parameterization of SNFS. *J. Math. Cryptol.*, 1(2):105–124, 2007.

[158] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 1993. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.

[159] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.

[160] Madhu Sudan. Ideal error-correcting codes: unifying algebraic and number-theoretic algorithms. In *Applied algebra, algebraic algorithms and error-correcting codes (Melbourne, 2001)*, volume 2227 of *Lecture Notes in Comput. Sci.*, pages 36–45. Springer, Berlin, 2001.

[161] James Joseph Sylvester. A method of determining by mere inspection the derivatives from two equations of any degree. *Philos. Mag.*, 16:132–135, 1840.

[162] James Joseph Sylvester. On the resultant of two congruences. *Johns Hopkins University Circulars*, 1:131, 1881.

[163] Emmanuel Thomé. Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. *J. Symbolic Comput.*, 33(5):757–775, 2002.

[164] Wilberd van der Kallen. Complexity of the Havas, Majewski, Matthews LLL Hermite normal form algorithm. *J. Symbolic Comput.*, 30(3):329–337, 2000.

[165] B. L. van der Waerden. *Modern Algebra. Vol. II.* Frederick Ungar Publishing Co., New York, N. Y., 1950. Translated from the second revised German edition by Theodore J. Benac.

[166] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra.* Cambridge University Press, New York, 1999.

[167] Ronnie S. Williams, Jr. Cubic polynomials in the number field sieve. Master's thesis, Texas Tech University, May 2010.

[168] Harald K. Wimmer. On the history of the Bezoutian and the resultant matrix. *Linear Algebra Appl.*, 128:27–34, 1990.

[169] J. M. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

Appendix A

Appendices for Chapter 5

A.1 Number Field Codes with Known Rate

Lenstra [106] and Guruswami [67] both use non-constructive shift parameters to obtain estimates on the rate of their number field codes. In this appendix, a family of number field codes with easily computable rate is presented. Like NF-codes, the new construction encodes elements of $\mathcal{O}_K$ by their residues modulo relatively prime ideals. However, a different "message set" is used:

Definition A.1.1. Let K be a number field that contains pairwise relatively prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_n \subseteq \mathcal{O}_K$, ordered so that $\mathfrak{N}\mathfrak{p}_1 \leq \mathfrak{N}\mathfrak{p}_2 \leq \ldots \leq \mathfrak{N}\mathfrak{p}_n$. Let R be a finite index additive subgroup of $\mathcal{O}_K$ and $\beta_1, \ldots, \beta_d$ be an integral basis for R. The code $\mathcal{C} = \mathcal{C}_K$, based on K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; \beta_1, \ldots, \beta_d)$, is defined to be the set

$$\mathcal{C} = \left\{ (m + \mathfrak{p}_1, \ldots, m + \mathfrak{p}_n) \in \frac{\mathcal{O}_K}{\mathfrak{p}_1} \times \ldots \times \frac{\mathcal{O}_K}{\mathfrak{p}_n} \mid m \in \mathcal{O}_K \cap \sum_{i=1}^{d} [0, 1)\beta_i \right\}.$$

The set $\mathcal{M}_{\mathcal{C}} = \mathcal{O}_K \cap \sum_{i=1}^{d} [0, 1)\beta_i$ is referred to as the *message set* of $\mathcal{C}$.

There are many possible choices of R in the code construction. For example, taking $R = \prod_{i=1}^{k} \mathfrak{p}_i$ for some $k \leq n$ leads to another generalisation of CRT codes to number fields. For this choice of R, the cardinality of the message set is $|\mathcal{M}_{\mathcal{C}}| = \prod_{i=1}^{k} \mathfrak{N}\mathfrak{p}_i$. Another simple example is to take $R = s_1\omega_1\mathbb{Z} + \ldots + s_d\omega_d\mathbb{Z}$, where $\omega_1, \ldots, \omega_d$ is an integral basis for $\mathcal{O}_K$ and the $s_1, \ldots, s_d \in \mathbb{Z}$. For this example, the cardinality of the message set is simply $|\mathcal{M}_{\mathcal{C}}| = \prod_{i=1}^{d} |s_i|$.

Let $\delta_{\mathbb{R}} : K \to \mathbb{R}^d$ be the injective group homomorphism defined in Section 5.3. Then, geometrically, the number of elements in the message set of a code based on K, with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; \beta_1, \ldots, \beta_d)$, is equal to the number of elements in intersection of $\delta_{\mathbb{R}}(\mathcal{O}_K)$ with the fundamental domain of the sublattice generated by the basis $(\delta_{\mathbb{R}}(\beta_1), \ldots, \delta_{\mathbb{R}}(\beta_d))$. Using this observation, the rate of the code is easily computed:

Theorem A.1.2. *Let $\mathcal{C}$ be a code based on K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; \beta_1, \ldots, \beta_d)$, where $\beta_1, \ldots, \beta_d$ is an integral basis for R. Then the rate of $\mathcal{C}$ is*

$$R(\mathcal{C}) = \frac{\log[\mathcal{O}_K : R]}{\sum_{i=1}^n \log \mathfrak{N}\mathfrak{p}_i} = \frac{\log\left(\det \delta_{\mathbb{R}}(R)\right) - \log \sqrt{|D_K|}}{\sum_{i=1}^n \log \mathfrak{N}\mathfrak{p}_i}.$$

Proof. Since $[\mathcal{O}_K : R]$ is finite, R is a full-rank sublattice of $\mathcal{O}_K$. Hence,

$$|\mathcal{M}_{\mathcal{C}}| = \left|\mathcal{O}_K \cap \sum_{i=1}^d [0,1)\beta_i\right| = [\mathcal{O}_K : R] = \frac{\det \delta_{\mathbb{R}}(R)}{\det \delta_{\mathbb{R}}(\mathcal{O}_K)} = \frac{\det \delta_{\mathbb{R}}(R)}{\sqrt{|D_K|}}. \qquad \square$$

Returning to the example where $R = \prod_{i=1}^k \mathfrak{p}_i$, for some $k \leq n$, the rate of the code is

$$R(\mathcal{C}) = \frac{\sum_{i=1}^k \log \mathfrak{N}\mathfrak{p}_i}{\sum_{i=1}^n \log \mathfrak{N}\mathfrak{p}_i} \geq \frac{k}{n} \cdot \frac{\log \mathfrak{N}\mathfrak{p}_1}{\log \mathfrak{N}\mathfrak{p}_n}.$$

For the special case where $\mathfrak{N}\mathfrak{p}_1 = \mathfrak{N}\mathfrak{p}_2 = \ldots = \mathfrak{N}\mathfrak{p}_n$, the rate of the code is simply $R(\mathcal{C}) = k/n$.

For a code that fits Definition A.1.1, the message set is dependent on the particular choice of the integral basis $\beta_1, \ldots, \beta_d$, whereas the rate of the code is not. However, the choice of the integral basis does influence the distance properties of the code. In particular, to construct codes with large distance, a basis for R that is short with respect to $T_2(x)$ (defined in Section 5.3) should be found:

Theorem A.1.3. *Let $\mathcal{C}$ be a code based on K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; \beta_1, \ldots, \beta_d)$, and define $M = \sum_{i=1}^d T_2(\beta_i)$. If $k \leq n$ satisfies $\prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i \geq M^{d/2}$, then the distance $d(\mathcal{C})$ of $\mathcal{C}$ is at least $(n - k + 1)$.*

Proof. Suppose $x, y \in \mathcal{M}_{\mathcal{C}}$ are distinct. Then there exist $x_1, \ldots, x_d, y_1, \ldots, y_d \in [0,1) \cap \mathbb{Q}$ such that $x = \sum_{i=1}^d x_i\beta_i$ and $y = \sum_{i=1}^d y_i\beta_i$. Therefore, applying the AM-GM and Cauchy-Schwarz inequalities, it follows that

$$\begin{aligned} |N_K(x-y)|^{\frac{2}{d}} &\leq \frac{1}{d}\sum_{j=1}^d \left|\sum_{i=1}^d (x_i - y_i)\sigma_j(\beta_i)\right|^2 \leq \frac{1}{d}\sum_{j=1}^d \left(\sum_{i=1}^d |x_i - y_i||\sigma_j(\beta_i)|\right)^2 \\ &\leq \frac{1}{d}\left(\sum_{i=1}^d |x_i - y_i|^2\right)\left(\sum_{j=1}^d\sum_{i=1}^d |\sigma_j(\beta_i)|^2\right) < \sum_{i=1}^d T_2(\beta_i). \end{aligned}$$

Therefore, if $\prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i \geq M^{d/2}$, for some $k \leq n$, then

$$\prod_{i=1}^n \mathfrak{N}\mathfrak{p}_i^{\sigma(x-y,\mathfrak{p}_i)} \leq |N_K(x-y)| < M^{\frac{d}{2}} \leq \prod_{i=1}^k \mathfrak{N}\mathfrak{p}_i.$$

Hence, $\sum_{i=1}^n \sigma(x - y, \mathfrak{p}_i) < k$. Since x and y were arbitrary distinct elements of $\mathcal{M}_{\mathcal{C}}$, it follows that $d(\mathcal{C}) \geq n - k + 1$. $\square$

A.2 Decoding with Nonzero Shift Parameters

The weighted list decoding algorithm for NF-codes developed in Section 5.3 required that the shift parameter $\boldsymbol{s} = \mathbf{0}$. However, the algorithm may still be utilised in the presence of nonzero shift parameters. In particular, at a potential cost of a reduction in decoding performance, the following algorithm can be used:

Algorithm A.2.1.

Input: A code $\mathcal{C}$ based on a number field K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M, \boldsymbol{s})$, where the $\mathfrak{p}_i$ are given in the form $\mathfrak{p}_i = \langle \alpha_i, \beta_i \rangle$ with $\alpha_i \neq 0$, for $1 \leq i \leq n$; a vector $(r_1 + \mathfrak{p}_1, \ldots, r_n + \mathfrak{p}_n) \in \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n$; an integral basis $\omega_1, \ldots, \omega_d$ for $\mathcal{O}_K$; and positive integers $z_1, \ldots, z_n$ and l.

Output: All $m \in \mathcal{M}_{\mathcal{C}}$ such that $\sum_{i=1}^n \sigma(m - r_i, \mathfrak{p}_i) z_i \log \mathfrak{N}\mathfrak{p}_i$ is sufficiently large.

1. Let $\boldsymbol{s} = (s_1, \ldots, s_{r_1+r_2})$ and define
$$\boldsymbol{s}' = \left(s_1, \ldots, s_{r_1}, \sqrt{2}\mathrm{Re}(s_{r_1+1}), \ldots, \sqrt{2}\mathrm{Re}(s_{r_1+r_2}), \sqrt{2}\mathrm{Im}(s_{r_1+1}), \ldots, \sqrt{2}\mathrm{Im}(s_{r_1+r_2})\right).$$
Using an algorithm for finding an approximate closest vector in a lattice (see [5, 88] and references therein), find a vector $\boldsymbol{y}' \in \delta_{\mathbb{R}}(\mathcal{O}_K)$ such that $\|\boldsymbol{s}' - \boldsymbol{y}'\|_2$ is small.
2. Set $\Delta = \sqrt{d}\,\|\boldsymbol{s}' - \boldsymbol{y}'\|_2$ and $y = \delta_{\mathbb{R}}^{-1}(\boldsymbol{y}')$. Apply Algorithm 5.3.2 to the NF-code $\mathcal{C}'$, based on K with parameters $(n, \mathfrak{p}_1, \ldots, \mathfrak{p}_n; M + \Delta, \mathbf{0})$, and the vector $((r_1 - y) + \mathfrak{p}_1, \ldots, (r_n - y) + \mathfrak{p}_n)$.
3. Return all $m' + y \in \mathcal{M}_{\mathcal{C}}$ such that $m' \in \mathcal{M}'_{\mathcal{C}}$ is an output of Algorithm 5.3.2 found in Step 2.

The following theorem provides a sufficient condition for an element $m \in \mathcal{M}_{\mathcal{C}}$ to be returned by Algorithm A.2.1:

Theorem A.2.2. *Algorithm A.2.1 returns all $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\prod_{i=1}^n \mathfrak{N}\mathfrak{p}_i^{\sigma(m-r_i,\mathfrak{p}_i)z_i} > 2^{\frac{d^2(l+1)-d}{4}} d^{-d} (l+1)^{\frac{d}{2}} (M+\Delta)^{\frac{dl}{2}} \sqrt{|D_K|} \left(\prod_{i=1}^n \mathfrak{N}\mathfrak{p}_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{l+1}}. \tag{A.1}$$

Proof. Define $\boldsymbol{y} = (\sigma_1(y), \ldots, \sigma_{r_1+r_2}(y)) \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$. Then

$$\mathrm{size}_{\boldsymbol{y}}(x) \leq \mathrm{size}_{\boldsymbol{s}}(x) + \mathrm{size}_{\boldsymbol{s}}(y) \leq \mathrm{size}_{\boldsymbol{s}}(x) + \sqrt{d}\left\|\boldsymbol{s}' - \boldsymbol{y}'\right\|_2, \quad \text{for all } x \in K.$$

Moreover, $\mathrm{size}_{\boldsymbol{y}}(x) = \mathrm{size}(x - y)$, for all $x \in K$. It follows that

$$\mathcal{M}_{\mathcal{C}} \subseteq \{m \in \mathcal{O}_K \mid \mathrm{size}_{\boldsymbol{y}}(m) \leq M + \Delta\} = \left\{m' + y \mid m' \in \mathcal{M}_{\mathcal{C}'}\right\}.$$

Therefore, given an element $m \in \mathcal{M}_{\mathcal{C}}$, there exists a corresponding element $m' \in \mathcal{M}_{\mathcal{C}'}$ such that $m = m' + y$ and $\sigma(m - r_i, \mathfrak{p}_i) = \sigma(m' - (r_i - y), \mathfrak{p}_i)$, for $1 \leq i \leq n$. Hence, if $m \in \mathcal{M}_{\mathcal{C}}$ satisfies (A.1), then Theorem 5.3.6 implies that the corresponding element $m' \in \mathcal{M}_{C'}$ is found by Algorithm 5.3.2 in Step 2, thus $m = m' + y$ is returned in Step 3. □

Algorithm A.2.1 reduces to the case where $\boldsymbol{s} = \mathbf{0}$ so that Algorithm 5.3.2 can be applied. However, the reduction comes at the cost of Algorithm 5.3.2 being applied with a potentially larger size bound ($M + \Delta$ compared to M). Therefore, it is important that Δ be as small as possible. Unfortunately, if K is a number field with large discriminant, then the existence of a vector $\boldsymbol{y}' \in \delta_{\mathbb{R}}(\mathcal{O}_K)$ such that $\|\boldsymbol{s}' - \boldsymbol{y}'\|_2$ is small, is not guaranteed in general.