An Introduction to the General Number Field Sieve

Matthew E. Briggs

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Dr. Ezra Brown, Chair
Dr. Martin Day
Dr. Charles Parry

April 17, 1998
Blacksburg, Virginia

Keywords: Number Field Sieve, Factoring, Cryptography, Algebraic Number Theory
Copyright 1998, Matthew E. Briggs

An Introduction to the General Number Field Sieve

Matthew E. Briggs

(ABSTRACT)

With the proliferation of computers into homes and businesses and the explosive growth rate of the Internet, the ability to conduct secure electronic communications and transactions has become an issue of vital concern. One of the most prominent systems for securing electronic information, known as RSA, relies upon the fact that it is computationally difficult to factor a "large" integer into its component prime integers. If an efficient algorithm is developed that can factor any arbitrarily large integer in a "reasonable" amount of time, the security value of the RSA system would be nullified.

The General Number Field Sieve algorithm is the fastest known method for factoring large integers. Research and development of this algorithm within the past five years has facilitated factorizations of integers that were once speculated to require thousands of years of supercomputer time to accomplish. While this method has many unexplored features that merit further research, the complexity of the algorithm prevents almost anyone but an expert from investigating its behavior. We address this concern by first pulling together much of the background information necessary to understand the concepts that are central in the General Number Field Sieve. These concepts are woven together into a cohesive presentation that details each theory while clearly describing how a particular theory fits into the algorithm. Formal proofs from existing literature are recast and illuminated to clarify their inner-workings and the role they play in the whole process. We also present a complete, detailed example of a factorization achieved with the General Number Field Sieve in order to concretize the concepts that are outlined.

Contents

List of Tables

Chapter 1

Introduction

The General Number Field Sieve (GNFS) is the fastest known method for factoring "large" integers, where large is generally taken to mean over 110 digits. This makes it the best algorithm for attempting to unscramble keys in the RSA [2, Chapter 4] public-key cryptography system, one of the most prevalent methods for transmitting and receiving secret data. In fact, GNFS was used recently to factor a 130-digit "challenge" number published by RSA, the largest number of cryptographic significance ever factored.

A specialized version of GNFS, the so-called "special" Number Field Sieve (SNFS), also exists; it is asymptotically faster than GNFS for factoring integers expressible in the form $r^e \pm s$ with $r, e, s \in \mathbb{Z}$ and $e > 0$. This has made SNFS the method of choice for attacking and successfully factoring the 155-digit ninth Fermat number [18], $2^{2^9} + 1$, as well as for factoring Mersenne "primes" and numbers on the Cunningham List [3], the latter being one of the oldest gauges of factoring technology.

Beyond its practical value, GNFS is also academically interesting. The algorithm itself uses ideas and results from diverse fields of mathematics and computer science. Algebraic number theory, graph theory, finite fields, linear algebra, and even real and complex analysis all play vital roles in GNFS.

The goal here is to describe the basic GNFS algorithm, explaining the relevant background information and theory the reader will need in order to understand the various stages of GNFS. We'll spend considerable time on an extended example, worked out in full detail so as to provide the reader a clear grasp of the previously outlined concepts. Once the fundamentals have been laid, we'll describe practical use of GNFS. This includes details on tuning GNFS for specific situations, as well as some of the general enhancements made to the base algorithm that improve its performance.

1.1 Cryptography and Factoring

Most cryptography systems make use of “one-way” functions, which intuitively can be thought of as mappings that are difficult to invert. In the RSA system [2, Chapter 4] the one-way function is multiplication of large prime integers, where large usually means over 110 digits. The key is that while multiplication of such integers can be done nearly instantly, the inversion function of factoring back into primes is virtually impossible.

In many of these systems, including the RSA procedure, an individual who wants the ability to receive encrypted messages that only he can read chooses a “public key” and a “private key.” The public key is available to anyone and is used to encrypt messages that can only be decrypted by someone who knows the corresponding private key. RSA facilitates this concept by having the private key consist of a set of three integers p, q, and d. The integers p and q are chosen to be large primes such that their product $n = p \cdot q$ is difficult to factor, while d is chosen relatively prime to $\phi(n)$, where $\phi(n)$ denotes Euler’s totient function for the number of integers less than or equal to n and relatively prime to n. The public key is comprised of the integer n and the integer e for which $d \cdot e \equiv 1 \pmod{\phi(n)}$.

To encrypt a message using the integers n and e for a public key, first encode the message [2, Chapter 4] as an integer M relatively prime to n. Let E denote the encrypted version of the message, where E is defined as

$$E = M^e \pmod{n}.$$

This integer E can be made available to anyone but it can only be decrypted back into the original message M by someone knowing the corresponding private key.

To decrypt the message, recall Euler’s formula [24, Theorem 2.8] which says that for any integers m and a with $\gcd(a, m) = 1$ that $a^{\phi(m)} \equiv 1 \pmod{m}$. Since M was chosen relatively prime to n it follows then that $M^{\phi(n)} \equiv 1 \pmod{n}$. Using this information leads to a method for decrypting E, since

$$E^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\cdot\phi(n)} \equiv M \cdot (M^{\phi(n)})^k \equiv M \pmod{n}$$

where $ed = 1 + k \cdot \phi(n)$ for some integer k since $d \cdot e \equiv 1 \pmod{\phi(n)}$ by construction. Note this computation recovers the original message M and makes use of information in the private key set.

To make clear the importance of factoring n, note that to decrypt the message the private key d was needed. Now e and n are in the public key set, and it is known that $d \cdot e \equiv 1 \pmod{\phi(n)}$, so d can be computed by computing the multiplicative inverse of e modulo $\phi(n)$. The latter is trivial, *assuming* $\phi(n)$ *is known.* Now from [24, Theorem 2.19]

$$\phi(n) = \phi(pq) = \phi(p) \cdot \phi(q) = (p - 1) \cdot (q - 1)$$

which is not immediately computable unless p and q are available. The latter is not a problem for someone with knowledge of the private key set, but for anyone else it entails factoring

n. Furthermore, if a method for computing $\phi(n)$ without knowing p or q is discovered, then n can be factored immediately. So the problem of unearthing the private key d boils down to computing $\phi(n)$, which in turn is equivalent to factoring n. If a method is discovered for factoring arbitrary integers quickly, then any RSA private key could be discovered and the system would become insecure.

1.2 Modern Methods of Factoring

The most straightforward method of factoring is trial division, where one simply tries to divide by each prime up to the square root of the number to factor. This method is indeed guaranteed to find a factor of any composite integer, but it is also guaranteed to be computationally infeasible for large enough integers.

To see why this is the case, suppose a 60-digit integer n is to be factored. Then one must check if n is divisible by any of the primes of size up to about 10^{30}. If the optimistic assumption is made that only 0.1% of these integers are prime, that still means about 10^{27} divisions need to take place. Again, being optimistic and assuming that enough computer resources are available to do 10^{15} of these divisions every second, it would still take roughly 10^{12} seconds, or over $31,000$ years to perform the computation. Of course, the algorithm might find a factor of n without extending all the way to the square root of n, so it may very well take a few thousand years less. On the other hand, the assumptions made were fairly optimistic to begin with so the algorithm would probably take longer than this rough projection. In any event, this obviously is not a practical method for factoring with run-times in the thousands of years!

Many of the successful factoring methods of the past twenty years have used the same basic technique, which itself dates back to the time of Fermat [26, 2]. The "difference of squares" method relies upon the observation that if integers x and y are such that $x \not\equiv y \pmod{n}$ and

$$x^2 \equiv y^2 \pmod{n} \tag{1.1}$$

then $\gcd(x-y,n)$ and $\gcd(x+y,n)$ are non-trivial factors of n.

If one is able to produce random integers x and y that satisfy (1.1), how likely is it that $\gcd(x+y,n)$ or $\gcd(x-y,n)$ is a non-trivial factor of n? In the case where n is the product of two distinct primes p and q, such as when n is a modulus used in the RSA method of §1.1, it turns out that a non-trivial factor of n is extracted in 2/3 of the cases, as seen in Table 1.1.

The question then becomes one of devising a means for producing integers x and y satisfying (1.1). The "random squares" algorithm of Dixon is one such method, which is not only of historical interest, but is also useful because it introduces concepts employed in the GNFS. Specifically, the notions of a *factor base* and being *smooth* over a factor base are introduced:

Table 1.1: Scenarios for $x^2 \equiv y^2 \pmod{n}$

$p \mid x+y$?	$p \mid x-y$?	$q \mid x+y$?	$q \mid x-y$?	$\gcd(x+y,n)$	$\gcd(x-y,n)$	Gives Factor?
Yes	Yes	Yes	Yes	n	n	
Yes	Yes	Yes	No	n	p	✓
Yes	Yes	No	Yes	p	n	✓
Yes	No	Yes	Yes	n	q	✓
Yes	No	Yes	No	n	1	
Yes	No	No	Yes	p	q	✓
No	Yes	Yes	Yes	q	n	✓
No	Yes	Yes	No	q	p	✓
No	Yes	No	Yes	1	n	

Definition 1.2.1. A nonempty set F of positive prime integers is called a *factor base.* An integer k is said to be *smooth* over the factor base F if all primes occurring in the unique factorization of k into primes are members of F.

The method of Dixon [2, pages 102–104] begins by fixing a factor base $F = \{p_1, p_2, \ldots, p_m\}$ and then proceeds to compute a set of random integers r_i with the property that $f(r_i) = r_i^2 \pmod{n}$ is smooth over F. When more than m such integers are found, a subset U of the integers in the sequence can be found such that

$$\prod_{r_i \in U} f(r_i) = p_1^{2e_1} p_2^{2e_2} \cdots p_m^{2e_m} = (p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m})^2$$

with $e_i \geq 0$. This set U is the key to producing a difference of squares, for if

$$x = \prod_{r_i \in U} r_i \quad \text{and} \quad y = p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$$

then a difference of squares follows from

$$x^2 = \prod_{r_i \in U} r_i^2 \equiv \prod_{r_i \in U} f(r_i) \equiv y^2 \pmod{n}.$$

Finding the set U turns out to be a reasonably straightforward task. For each $r_i \in U$ one can associate a vector $v_i \in \mathbb{F}_2^m$, where $\mathbb{F}_2^m$ denotes the m-dimensional vector space over the finite field $\mathbb{Z}/2\mathbb{Z}$ of 2 elements. The j^{th} coordinate of v_i is set to 0 if the prime p_j divides $f(r_i)$ an even number of times and is set to 1 otherwise. It's a standard result from linear algebra [11, Theorem 1.10] that if more than m such vectors are collected then there is a non-trivial linear dependence among them. In this particular case, that means a nonempty set of v_i vectors can be produced whose sum yields the zero vector. But since these vectors represent the parity of the exponents of the primes that occur in the factorization of the $f(r_i)$'s, it follows that the product of the $f(r_i)$ values corresponding to the v_i's that occur in a dependency is a perfect square. The set U can then be constructed from the r_i whose vector v_i occurs in a dependency. Many well-studied and efficient techniques exist for finding dependencies among vectors, such as Gaussian elimination [2, pages 114–115]. The real question then becomes one of finding enough r_i with $f(r_i)$ smooth over F, and doing so in a timely fashion..

1.3 The Quadratic Sieve

The Quadratic Sieve (QS) factoring algorithm of Carl Pomerance [26, 2] was the most effective general-purpose factoring algorithm of the 1980's and the early 90's, and is still the method of choice for integers between 50 and 100 digits. At its heart the QS is essentially Dixon's algorithm, in the sense that it uses factor bases, smoothness, and dependencies among vectors over $\mathbb{Z}/2\mathbb{Z}$ to produce its squares. Through a slight modification of the polynomials $f(x)$ considered, however, QS sets itself apart from Dixon's method by finding smooth values in a remarkably fast manner.

As with Dixon's method, QS begins by fixing a factor base $F = \{p_1, p_2, \ldots, p_m\}$. Instead of searching for integers r_i for which $f(r_i) = r_i^2 \pmod{n}$ is smooth over F, values of $f(r_i) = r_i^2 - n$ are sought which are smooth over F. Again, as is the case in Dixon's method, when more than m such integers are found, a subset U can be produced with

$$\prod_{r_i \in U} f(r_i) = p_1^{2e_1} p_2^{2e_2} \cdots p_m^{2e_m} = (p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m})^2.$$

A difference of squares follows by letting

$$x = \prod_{r_i \in U} r_i \quad \text{and} \quad y = p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m}$$

since then

$$x^2 = \prod_{r_i \in U} r_i^2 \equiv \prod_{r_i \in U} (r_i^2 - n) \equiv \prod_{r_i \in U} f(r_i) \equiv y^2 \pmod{n}. \tag{1.2}$$

At this point the QS does not look that much different from Dixon's algorithm, and in reality it is not. The only difference is in the polynomial $f(r_i) = r_i^2 - n$ used, and in fact it is the special form of this polynomial that allows the dramatic increase in speed alluded to earlier.

The big improvement comes in how the different r_i are chosen when considering whether $f(r_i)$ is smooth over F. The straightforward approach is to pick a random integer r_i and then to trial-divide $f(r_i)$ by the primes in F. If $f(r_i)$ factors completely over F then r_i is added to the set of "useful" r_i, otherwise it is discarded. In either case, a new random r_i is picked and the process continues until more than m integers exist in the set of useful r_i values.

The problem with this approach is that a lot of time is wasted attempting to divide by primes in F that don't evenly divide a particular $f(r_i)$. A dramatic improvement can be made by changing the focus of the operations. Instead of concentrating on a fixed $f(r_i)$ and finding out which primes in the factor base divide it, fix a prime $p \in F$ and determine which $f(r_i)$ values are divisible by that p. If determining the $f(r_i)$ values that are divisible by a fixed prime $p \in F$ can be done in a reasonable manner, then one saves the time of attempting to divide by primes that don't divide into an $f(r_i)$.

To see how the special form of $f(r_i) = r_i^2 - n$ facilitates this, fix an r_i and a prime $p \in F$ and suppose p divides $f(r_i)$. Then $f(r_i) \equiv 0 \pmod{p}$ and hence $r_i^2 - n \equiv 0 \pmod{p}$ by the definition of the function f. Then for any integer k it follows that

$$f(r_i + kp) = r_i^2 + 2r_ikp + k^2p^2 - n \equiv r_i^2 - n \equiv 0 \pmod{p}$$

and hence p divides $f(r_i + kp)$ as well. Thus, the real work in finding values of r_i for which $f(r_i)$ is divisible by p amounts to initially solving the quadratic congruence $r_i^2 \equiv n \pmod{p}$, for which there is an easy and efficient method [16, Section 9.2]. The rest of the r_i values for which $f(r_i)$ is divisible by p are then $r_i + pk$ for $k \in \mathbb{Z}$. In order for this procedure to work, n must be a quadratic residue modulo p, hence F shouldn't contain any primes for which n is a quadratic non-residue.

In practice one selects a bound $-u < r_i < u$ on the r_i values for which it is expected more than m of the possible values of $f(r_i)$ within this range will be smooth over F. An array of computer memory is then initialized to the $f(r_i)$ values within this range. For each prime $p \in F$, the quadratic congruence $r_i^2 \equiv n \pmod{p}$ is solved. Then for all integers k for which $-u < r_i + pk < u$, the prime p is divided out of the corresponding $f(r_i + pk)$ value. Finally, after this procedure has been performed for every prime p, the array of computer memory is scanned for values of $f(r_i)$ for which $f(r_i) = 1$. These correspond to values of r_i whose $f(r_i)$ factor completely over the factor base.

Using this sieving technique, every division by a prime p is "useful" in the sense that it is always guaranteed to divide into the $f(r_i)$ value that is selected, which is not the case at all with blind trial division. Since division is one of the most time-consuming operations in a computer, this shift in focus leads to a dramatic speed-up.

Chapter 2

Motivation for the General Number Field Sieve

Almost all difference of squares methods that produce integers x and y as in (1.1) use the same basic concepts of a factor base, smoothness, and finding dependencies among vectors over $\mathbb{Z}/2\mathbb{Z}$. Key breakthroughs occur when new methods are developed or older methods enhanced to produce more smooth values over a factor base in a time dramatically less than previous methods. For instance, Dixon's method is improved upon by QS by adjusting the polynomials that are used, and more importantly by changing the perspective on how smooth values are searched for. This latter shift in perspective facilitates a fast sieving procedure for finding smooth values, and hence a breakthrough in factoring technology.

The ideas leading to the GNFS algorithm are motivated by similar techniques that led to the development of the QS from Dixon's method. As expected, then, the notions of a factor base, smoothness and dependency-finding are used in the GNFS, along with a perspective for finding smooth values that supports a sieving procedure. A big break through comes by first realizing that the quadratic polynomials of Dixon's method and the QS don't necessarily have to be quadratic. Perhaps certain cubic, quartic, quintic, or even higher degree polynomials could produce more smooth values than quadratics.

Another less obvious improvement stems from somehow allowing other rings besides $\mathbb{Z}$ and $\mathbb{Z}/n\mathbb{Z}$ into the algorithm. The idea here is that other rings could potentially have a notion of smoothness imposed on them, similar to the notion over $\mathbb{Z}$, with the hopes that more smooth values exist in such rings than in $\mathbb{Z}$. Furthermore, if some kind natural mapping existed between such rings and $\mathbb{Z}/n\mathbb{Z}$, then a way of producing a difference of squares could possibly be arrived at.

This idea of using other rings to produce a difference of squares is explored in §2.1 and then tied in with higher degree polynomials in §2.2, §2.3, and §2.4.

2.1 Generalizing the Quadratic Sieve

To see how rings other than $\mathbb{Z}$ and $\mathbb{Z}/n\mathbb{Z}$ can come into play in a difference of squares method, one only has to generalize the role the polynomial $f(r_i) = r_i^2 - n$ plays in the QS. Recall from §1.3 that in the QS a set of integers U is found such that (1.2) holds. Then the polynomial $f(r_i) = r_i^2 - n$ can be thought of as a ring homomorphism $f : \mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}$. In particular, f maps the product of all $f(r_i)$ for $r_i \in U$, which is smooth over the factor base F (by the choice of the r_i) and a perfect square in $\mathbb{Z}$ (by the definition of U), to a perfect square in $\mathbb{Z}/n\mathbb{Z}$. The important point is that f maps a square in the ring $\mathbb{Z}$ to a square in the ring $\mathbb{Z}/n\mathbb{Z}$, which supplies the integers x and y for (1.1).

Suppose that there exists a ring R and a ring homomorphism $\phi : R \to \mathbb{Z}/n\mathbb{Z}$. If $\beta \in R$ with $\phi(\beta^2) = y^2 \pmod{n}$ and $x = \phi(\beta) \pmod{n}$ then

$$x^2 \equiv \phi(\beta)^2 \equiv \phi(\beta^2) \equiv y^2 \pmod{n}.$$

So if an element in R can be found that is a perfect square in R and which maps to a perfect square in $\mathbb{Z}/n\mathbb{Z}$, then applying ϕ will yield a difference of squares. As will be seen in the following sections, there is a natural way to construct such rings and the corresponding homomorphisms to $\mathbb{Z}/n\mathbb{Z}$ that will yield a difference of squares.

2.2 Fields and Roots of Irreducible Polynomials

Suppose a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients is known. Then $f(x)$ splits into distinct linear factors over the complex numbers [10, Section 13.4] as

$$f(x) = (x - \theta_1)(x - \theta_2) \cdots (x - \theta_d)$$

with $\theta_i \in \mathbb{C}$. One can choose any root $\theta = \theta_i$ and form a ring in a manner that is easy to verify [14, Chapter 5, Theorem 1.3]:

Proposition 2.2.1. *If θ denotes a complex root of a monic, irreducible polynomial $f(x)$ with rational coefficients, then the set of all polynomials in θ with rational coefficients, denoted $\mathbb{Q}(\theta)$, forms a ring.*

In fact, much more is true of $\mathbb{Q}(\theta)$ because of the monic, irreducible nature of the defining polynomial $f(x)$:

Theorem 2.2.2. *Given a monic, irreducible polynomial $f(x)$ with rational coefficients, a root $\theta \in \mathbb{C}$ of $f(x)$, and the associated ring $\mathbb{Q}(\theta)$, the following hold:*

1. *$\mathbb{Q}(\theta) \cong \mathbb{Q}[x]/(f(x))$.*

2. *$\mathbb{Q}(\theta)$ is a field.*

3. *$f(x)$ divides any polynomial $g(x)$ for which $g(\theta) = 0$.*

4. *The set $\{1, \theta, \theta^2, \ldots, \theta^{d-1}\}$ forms a basis for $\mathbb{Q}(\theta)$ as a vector space over $\mathbb{Q}$.*

Proof. Proceeding along the lines of [14, Chapter 5, Theorem 1.6], let $\phi : \mathbb{Q}[x] \to \mathbb{Q}(\theta)$ denote the natural map with $\phi(\mathbb{Q}) = \mathbb{Q}$ and $\phi(x) = \theta$, which maps polynomials in x to polynomials in θ. It is clear that this mapping is actually a surjective ring homomorphism. Now $f(x)$ maps to $f(\theta) = 0$ under the mapping ϕ so that $f(x) \in \ker \phi$. In fact, since $\mathbb{Q}$ is a field it follows that $\mathbb{Q}[x]$ is a principal ideal domain [14, Chapter 3, Theorem 3.9] and hence $\ker \phi = (g(x))$ for some polynomial $g(x)$. Now $f(x) \in \ker \phi = (g(x))$ implies that $f(x)$ is a multiple of $g(x)$, and since $f(x)$ is irreducible is follows that $g(x)$ and $f(x)$ must differ by at most a scalar so that $\ker \phi = (g(x)) = (f(x))$. Now $\mathbb{Q}[x]/\ker \phi \cong \operatorname{Im} \phi$ and since ϕ is onto it follows that $\mathbb{Q}[x]/(f(x)) \cong \mathbb{Q}(\theta)$ and the first part of the theorem follows.

To prove the second condition, note $(0) \subsetneqq (f(x)) \subsetneqq \mathbb{Q}[x]$ since $f(x)$ is not identically 0 and $\ker \phi = (f(x))$ is not the whole ring $\mathbb{Q}[x]$ because ϕ maps the non-zero rationals to non-zero rationals. Since $f(x)$ is irreducible it follows [29, Proposition 4.4] that $(f(x))$ is maximal with respect to all proper principal ideals of $\mathbb{Q}[x]$. But $\mathbb{Q}[x]$ is a principal ideal domain and hence $(f(x))$ is a maximal ideal of $\mathbb{Q}[x]$. Thus $\mathbb{Q}[x]/(f(x)) \cong \mathbb{Q}(\theta)$ is a field [29, Lemma 5.1].

For the third part, consider a polynomial $g(x)$ for which $g(\theta) = 0$. Then $\phi(g(x)) = g(\theta) = 0$ implies that $g(x) \in \ker \phi$ and since $\ker \phi = (f(x))$ it follows that $g(x)$ is a multiple of $f(x)$.

Finally, to prove the result about a representation for $\mathbb{Q}(\theta)$ as a vector space over $\mathbb{Q}$, let $g(\theta) \in \mathbb{Q}(\theta)$ be any polynomial in θ and $g(x) \in \mathbb{Q}[x]$ its corresponding representation as a polynomial in x. By the division algorithm, $g(x)$ may be written as

$$g(x) = q(x) \cdot f(x) + r(x)$$

where $\deg r(x) < \deg f(x)$. Then

$$g(\theta) = q(\theta) \cdot f(\theta) + r(\theta) = q(\theta) \cdot 0 + r(\theta) = r(\theta)$$

since θ is a root of $f(x)$. It follows then that $g(\theta)$ may be written in the form

$$g(\theta) = a_{d-1}\theta^{d-1} + a_{d-2}\theta^{d-2} + a_1\theta + a_0$$

where the $a_i \in \mathbb{Q}$ are the coefficients of $r(x)$, since $r(x)$ is a polynomial of degree strictly less than d. Thus the set $\{1, \theta, \theta^2, \ldots, \theta^{d-1}\}$ spans $\mathbb{Q}(\theta)$ as a vector space over $\mathbb{Q}$. To show linear independence, suppose that $a_{d-1}\theta^{d-1} + a_{d-2}\theta^{d-2} + \cdots + a_1\theta + a_0 = 0$. Letting $g(x)$ be the polynomial

$$g(x) = a_{d-1}x^{d-1} + a_{d-2}x^{d-2} + \cdots + a_1x + a_0$$

it follows by construction that $g(\theta) = 0$. Thus, $g(x) \in \ker \phi = (f(x))$ and hence $f(x)$ must divide $g(x)$. But the degree of $f(x)$ is strictly greater than $g(x)$ so that $g(x)$ must be the zero polynomial and hence $a_i = 0$ for all $0 \leq i < d$ and linear independence follows. □

2.3 Rings of Algebraic Integers

At this point, to summarize the important developments, we have shown that a monic, irreducible polynomial $f(x)$ of arbitrary degree d and with rational coefficients gives rise to a field $\mathbb{Q}(\theta)$ where $\theta \in \mathbb{C}$ is a root of $f(x)$. Furthermore, elements of $\mathbb{Q}(\theta)$ can be conveniently represented as $\mathbb{Q}$-linear combinations of the elements $S = \{1, \theta, \theta^2, \ldots, \theta^{d-1}\}$. Although the latter representation is convenient, working with $\mathbb{Z}$-linear combinations of the elements of S would be easier since then denominators could be discounted. Further analysis of the field $\mathbb{Q}(\theta)$ turns up a ring whose elements can be represented in just such a manner.

Definition 2.3.1. A complex number α is called an *algebraic integer* if it is the root of a monic polynomial with integer coefficients.

Thus, if $f(x)$ is an irreducible, monic polynomial of degree d with integer coefficients and $\theta \in \mathbb{C}$ is a root of $f(x)$, it follows that θ is an algebraic integer according to this definition. The following is a standard result from algebraic number theory that justifies the definition:

Proposition 2.3.1. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, the set of all algebraic integers in $\mathbb{Q}(\theta)$, denoted $\mathfrak{O}$, forms a subring of the field $\mathbb{Q}(\theta)$.*

The actual ring that will be used in the GNFS is subring of the ring of algebraic integers $\mathfrak{O}$ of $\mathbb{Q}(\theta)$ which possesses the convenient representation mentioned in the beginning paragraph:

Proposition 2.3.2. *Given a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, the set of all $\mathbb{Z}$-linear combinations of the elements $\{1, \theta, \theta^2, \ldots, \theta^{d-1}\}$, denoted $\mathbb{Z}[\theta]$, forms a subring of the ring of algebraic integers $\mathfrak{O}$ of $\mathbb{Q}(\theta)$.*

Before going further, it should be pointed out that the subring $\mathbb{Z}[\theta]$ can indeed be a proper subring of $\mathfrak{O}$. For instance, the polynomial $x^2 - 5$ is easily seen to be irreducible and monic so that $\mathbb{Q}\left(\sqrt{5}\right)$ forms a field, which is a vector space over $\mathbb{Q}$ with basis $S = \left\{1, \sqrt{5}\right\}$. If $\alpha = \left(1 + \sqrt{5}\right)/2$ then $\alpha \in \mathbb{Q}\left(\sqrt{5}\right)$ since α is a $\mathbb{Q}$-linear combination of the elements of S. Furthermore α satisfies the polynomial $g(x) = x^2 - x - 1$ and is hence an algebraic integer, but clearly $\alpha \notin \mathbb{Z}[\theta]$. Hence, $\mathbb{Q}\left(\sqrt{5}\right)$ possesses an algebraic integer which is not contained in $\mathbb{Z}\left[\sqrt{5}\right]$ and so

$$\mathbb{Z}\left[\sqrt{5}\right] \subsetneqq \mathfrak{O} \subsetneqq \mathbb{Q}\left(\sqrt{5}\right).$$

2.4 Producing a Difference of Squares

Having demonstrated that for any monic, irreducible polynomial an associated ring can be constructed that has a natural representation as $\mathbb{Z}$-linear combinations of elements from a finite set, the natural question is how to map that ring onto $\mathbb{Z}/n\mathbb{Z}$ so as to produce a difference of squares. The next proposition [5, page 53] addresses this issue:

Proposition 2.4.1. *Given a monic, irreducible polynomial $f(x)$ with integer coefficients, a root $\theta \in \mathbb{C}$ of $f(x)$, and an integer $m \in \mathbb{Z}/n\mathbb{Z}$ for which $f(m) \equiv 0 \pmod{n}$, the mapping $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/n\mathbb{Z}$ with $\phi(1) = 1 \pmod{n}$ and which sends θ to m is a surjective ring homomorphism.*

To see how this can result in a difference of squares, suppose a set U of pairs of integers (a, b) can be found such that

$$\prod_{(a,b)\in U} (a + b\theta) = \beta^2 \quad \text{and} \quad \prod_{(a,b)\in U} (a + bm) = y^2$$

with $\beta \in \mathbb{Z}[\theta]$ and $y \in \mathbb{Z}$. Then applying the natural homomorphism ϕ from Proposition 2.4.1 and letting $\phi(\beta) = x \in \mathbb{Z}/n\mathbb{Z}$, it follows that

$$\begin{aligned} x^2 \equiv \phi(\beta)^2 \equiv \phi(\beta^2) \equiv \phi\left(\prod_{(a,b)\in U} (a + b\theta)\right) \\ \equiv \prod_{(a,b)\in U} \phi(a + b\theta) \equiv \prod_{(a,b)\in U} (a + bm) \equiv y^2 \pmod{n} \end{aligned}$$

and a difference of squares results.

Note 2.4.1. The condition that the product of the elements $a + b\theta$ corresponding to pairs in U be a perfect square in $\mathbb{Z}[\theta]$ is imposed because the ring homomorphism ϕ is only defined on elements of $\mathbb{Z}[\theta]$. In practice the condition is relaxed to allow for the product being a perfect square in $\mathbb{Q}(\theta)$, which is less restrictive and hence more likely to be satisfied. Now if

$$\prod_{(a,b)\in U} (a + b\theta) = \alpha^2 \tag{2.1}$$

for some $\alpha \in \mathbb{Q}(\theta)$, it follows [5, pages 60–61] that $\alpha \in \mathfrak{O}$ and in fact $f'(\theta) \cdot \alpha \in \mathbb{Z}[\theta]$. A difference of squares can still be produced, for if

$$\prod_{(a,b)\in U} (a + b\theta) = \alpha^2 \quad \text{and} \quad \prod_{(a,b)\in U} (a + bm) = z^2 \tag{2.2}$$

with $\alpha \in \mathfrak{O}$ and $z \in \mathbb{Z}$, then letting $\beta = f'(\theta) \cdot \alpha \in \mathbb{Z}[\theta]$, $y = f'(m) \cdot z$, and $x = \phi(\beta) \in \mathbb{Z}/n\mathbb{Z}$, it follows that

$$\begin{aligned} x^2 \equiv \phi(\beta)^2 \equiv \phi(\beta^2) \equiv \phi\left(f'(\theta)^2 \cdot \prod_{(a,b)\in U} (a + b\theta)\right) \\ \equiv \phi\left(f'(\theta)\right)^2 \cdot \prod_{(a,b)\in U} \phi(a + b\theta) \equiv f'(m)^2 \cdot \prod_{(a,b)\in U} (a + bm) \equiv y^2 \pmod{n} \end{aligned}$$

and another difference of squares has been produced.

Chapter 3

The General Number Field Sieve Algorithm

In the quadratic sieve detailed in §1.3, a factor base F of prime integers is selected and a set U of integers is then found such that all $r_i \in U$ have $f(r_i)$ smooth over F and

$$\prod_{r_i \in U} f(r_i) = y^2$$

for some $y \in \mathbb{Z}$. Because of the special form of the polynomial $f(r_i) = r_i^2 - n$ it follows immediately that a perfect square in $\mathbb{Z}/n\mathbb{Z}$ is also produced since

$$\prod_{r_i \in U} f(r_i) \equiv \prod_{r_i \in U} (r_i^2 - n) \equiv \prod_{r_i \in U} r_i^2 \equiv \left(\prod_{r_i \in U} r_i \right)^2 \pmod{n}$$

and as shown in Table 1.1 there is a better than 50-50 chance that this will produce a non-trivial factor of n. The important point to notice is that the square in $\mathbb{Z}/n\mathbb{Z}$ comes for "free", in that for any set T of arbitrary integers, the product of the corresponding $f(r_i)$ for all $r_i \in T$ is a perfect square in $\mathbb{Z}/n\mathbb{Z}$ because of the form of $f(r_i) = r_i^2 - n$. When producing the squares in the QS then, finding one square essentially requires no work, while the other square is found through sieving and linear algebra.

The GNFS algorithm extends beyond quadratic polynomials to allow for any higher degree, so that a square is not automatically produced in $\mathbb{Z}/n\mathbb{Z}$ as it is in the QS. A natural technique for finding (a, b) pairs that satisfy (2.2) is to to combine a notion of smoothness in $\mathbb{Z}[\theta]$ with smoothness in $\mathbb{Z}$and to search for (a, b) pairs with $a + b\theta$ smooth over an "algebraic" factor base for $\mathbb{Z}[\theta]$ and $a + bm$ smooth over a "rational" factor base for $\mathbb{Z}$. As in QS, when enough pairs are found that are "simultaneously smooth" over the two factor bases, then hopefully a square in both $\mathbb{Z}[\theta]$ and $\mathbb{Z}$ can be produced according to (2.2). Indeed, this is exactly how the GNFS algorithm proceeds.

Generalizing the notion of a factor base to $\mathbb{Z}[\theta]$ and defining what it means to be smooth over such a factor base is discussed in §3.1. Producing squares in $\mathbb{Z}[\theta]$ using smoothness over a factor base is discussed in §3.1, §3.2, and §3.3. Sieving in $\mathbb{Z}$ and $\mathbb{Z}[\theta]$ is outlined in §3.4, §3.5, §3.6, §3.7, and §3.8.

3.1 Smoothness And The Algebraic Factor Base

Since a factor base over $\mathbb{Z}$ is simply a set of prime integers, the immediate analog for $\mathbb{Z}[\theta]$ would seem to be a set of distinct irreducible elements of the ring $\mathbb{Z}[\theta]$. Early implementations of the Special Number Field Sieve [19], a predecessor of the GNFS, actually did use such factor bases, but they also made the assumptions that $\mathbb{Z}[\theta] = \mathfrak{O}$ and $\mathbb{Z}[\theta]$ is a unique factorization domain, neither of which is true in general. Furthermore, even when these latter two assumptions were true, the resulting implementations were awkward and unwieldy because units of $\mathbb{Z}[\theta]$ had to also be added to the factor base and then figured into computations at later stages of the algorithm.

The solution turns out to be to maintain a factor base not of prime elements of $\mathbb{Z}[\theta]$ but rather of ideals of $\mathbb{Z}[\theta]$ of a special form (the special form eases the development of a sieving procedure). The following recalls a well-known fact [29, Theorem 5.3] from algebraic number theory that justifies the use of ideals in a factor base:

Proposition 3.1.1. *Given a monic, irreducible polynomial $f(x)$ of degree d with integers coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, then the ring of algebraic integers $\mathfrak{O}$ forms a Dedekind domain. In particular, this implies:*

1. *The ring $\mathfrak{O}$ is noetherian.*
2. *Prime ideals of $\mathfrak{O}$ are maximal ideals of $\mathfrak{O}$, and vice versa.*
3. *Using the canonical notion of ideal multiplication, ideals of $\mathfrak{O}$ can be uniquely factored, up to order, into prime ideals of $\mathfrak{O}$.*

The high-level idea then is to choose a set I of prime ideals of $\mathfrak{O}$, which will be called an *algebraic factor base*, and to find (a, b) pairs for which the element $a+b\theta$ has a principal ideal $\langle a + b\theta\rangle$ that factors completely into prime ideals of I. Such an element is said to be *smooth* over the algebraic factor base I. By collecting more (a, b) pairs than ideals in I, hopefully some of the $a + b\theta$ values corresponding these pairs can be multiplied together to produce a perfect square in $\mathbb{Z}[\theta]$, in a manner analogous to the QS.

To begin fleshing out this strategy, it is essential that the concept of the "norm" function be developed. This function, as it will turn out, allows for questions about factorization of elements in $\mathbb{Z}[\theta]$ and ideals of $\mathfrak{O}$ to be answered by addressing similar questions in $\mathbb{Z}$. Begin then with the following observation:

Theorem 3.1.2. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, there are exactly d ring monomorphisms (embeddings) from the field $\mathbb{Q}(\theta)$ into the field $\mathbb{C}$. These embeddings are given by $\sigma_i(\mathbb{Q}) = \mathbb{Q}$ and $\sigma_i(\theta) = \theta_i$ for $1 \leq i \leq d$, assuming $f(x)$ splits over $\mathbb{C}$ as*

$$f(x) = (x - \theta_1)(x - \theta_2) \cdots (x - \theta_d).$$

Proof. Note that the canonical mapping $\sigma_i : \mathbb{Q}(\theta) \to \mathbb{Q}(\theta_i)$ which sends θ to θ_i for $1 \leq i \leq d$ is an isomorphism of fields [14, Chapter 5, Corollary 1.9], so that each σ_i determines a distinct isomorphic copy of $\mathbb{Q}(\theta)$ in $\mathbb{C}$. Thus, there are at least d embeddings from $\mathbb{Q}(\theta)$ into $\mathbb{C}$.

To show that these σ_i are the only such embeddings, suppose $\sigma : \mathbb{Q}(\theta) \to \mathbb{C}$ is a ring monomorphism. Then in particular $\sigma(\mathbb{Q}) = \mathbb{Q}$. Now if $\sigma(\theta) = \alpha \in \mathbb{C}$ and $f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_1x + a_0$ then

$$\begin{aligned} f(\alpha) = \alpha^d + a_{d-1}\alpha^{d-1} + \cdots + a_1\alpha + a_0 &= \phi(\theta)^d + a_{d-1}\phi(\theta)^{d-1} + \cdots + a_1\phi(\theta) + a_0 \\ &= \phi(\theta^d + a_{d-1}\theta^{d-1} + \cdots + a_1\theta + a_0) = \phi(0) = 0 \end{aligned}$$

and hence $\alpha = \theta_i$ and $\sigma = \sigma_i$ for some $1 \leq i \leq d$. Thus, the σ_i are the only embeddings of $\mathbb{Q}(\theta)$ into $\mathbb{C}$and there exactly d of them. □

The embeddings of Theorem 3.1.2 allow for the definition of the norm function which maps elements of $\mathbb{Q}(\theta)$ to elements of $\mathbb{C}$:

Definition 3.1.1. Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients, a root $\theta \in \mathbb{C}$ of $f(x)$ and an element $\alpha \in \mathbb{Q}(\theta)$, the *norm* of the element α, denoted by $\mathrm{N}(\alpha)$, is defined as

$$\mathrm{N}(\alpha) = \sigma_1(\alpha)\sigma_2(\alpha) \cdots \sigma_d(\alpha) \tag{3.1}$$

where the σ_i are in the distinct embeddings of $\mathbb{Q}(\theta)$ into $\mathbb{C}$ as detailed in Theorem 3.1.2.

The real power of the norm function, as it is used in the GNFS, stems from the following standard result [29, pages 54–56] from algebraic number theory:

Proposition 3.1.3. *Given a monic, irreducible polynomial $f(x)$ of degree d with rational coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, the norm map of Definition 3.1.1 is a multiplicative function that maps elements of $\mathbb{Q}(\theta)$ to $\mathbb{Q} \subset \mathbb{C}$. Furthermore, algebraic integers in $\mathbb{Q}(\theta)$ are mapped to elements of $\mathbb{Z}$.*

Corollary. *Given a monic, irreducible polynomial $f(x)$ of degree d with integers coefficients and a root $\theta \in \mathbb{C}$ of $f(x)$, then the norm function of Definition 3.1.1 is a multiplicative function that sends elements of $\mathbb{Z}[\theta]$ to elements of $\mathbb{Z}$.*

Though Proposition 3.1.3 and its corollary are initially useful because they allow for recasting of questions about factorizations of elements of $\mathbb{Z}[\theta]$ to factorization in $\mathbb{Z}$, the full power of these results comes when the concept of the norm of an element is tied in with the norm of an ideal. Begin with the definition of the norm of an ideal:

Definition 3.1.2. Given a ring R and an ideal $\mathfrak{I}$ of R, the *norm* of $\mathfrak{I}$ is defined to be $[R : \mathfrak{I}]$, the number of cosets of $\mathfrak{I}$ in R.

The following results recalls elementary properties of the norm function on ideals of $\mathfrak{O}$, and explicitly relates the norm of an element of $\mathfrak{O}$ to the norm of the principal ideal generated by that element:

Proposition 3.1.4. *Let $f(x)$ be a monic, irreducible polynomial of degree d with rational coefficients and $\theta \in \mathbb{C}$ a root of $f(x)$. Then the norm function of Definition 3.1.2 is a multiplicative function that maps ideals of $\mathfrak{O}$ to positive integers. Moreover, if $\alpha \in \mathfrak{O}$ then* $\mathrm{N}(\langle\alpha\rangle) = |\,\mathrm{N}(\alpha)|$.

A final result [29, Theorem 5.11] from algebraic number theory clarifies how prime ideals of $\mathfrak{O}$ and prime integers are related:

Proposition 3.1.5. *Let D be a Dedekind domain. If $\mathfrak{p}$ is an ideal of D with* $\mathrm{N}(\mathfrak{p}) = p$ *for some prime integer p, then $\mathfrak{p}$ is a prime ideal of D. Conversely, if $\mathfrak{p}$ is a prime ideal of D then* $\mathrm{N}(\mathfrak{p}) = p^e$ *for some prime integer p and positive integer e.*

Given any element $\beta \in \mathfrak{O}$ it follows from Proposition 3.1.1 that the principal ideal $\langle\beta\rangle$ of $\mathfrak{O}$ factors uniquely as

$$\langle\beta\rangle = \mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\cdots\mathfrak{p}_k^{e_k}$$

for distinct prime ideals $\mathfrak{p}_i$ of $\mathfrak{O}$ and positive integers e_i with $1 \leq i \leq k$. Furthermore, Proposition 3.1.4 and Proposition 3.1.5 indicate that

$$\begin{aligned} |\,\mathrm{N}(\beta)| = \mathrm{N}(\langle\beta\rangle) = \mathrm{N}(\mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\cdots\mathfrak{p}_k^{e_k}) &= \mathrm{N}(\mathfrak{p}_1)^{e_1}\,\mathrm{N}(\mathfrak{p}_2)^{e_2}\cdots\mathrm{N}(\mathfrak{p}_k)^{e_k} \\ &= (p_1^{f_1})^{e_1}(p_2^{f_2})^{e_2}\cdots(p_k^{f_k})^{e_k} = p_1^{e_1+f_1}p_2^{e_2+f_2}\cdots p_k^{e_k+f_k} \end{aligned} \tag{3.2}$$

for (not necessarily distinct) primes p_i and positive integers e_i and f_i with $1 \leq i \leq k$. It is (3.2) that becomes the key tool for determining when an ideal $\langle a + b\theta\rangle$ factors completely over an algebraic factor base of prime ideals.

One very practical problem that presents itself is coming up with a representation for prime ideals that can easily be stored in a computer, and more importantly, that facilitates a sieving procedure for finding smooth $a + b\theta$ values. This is accomplished in GNFS by restricting the algebraic factor base to prime ideals of $\mathbb{Z}[\theta]$ of a special form instead of prime ideals of $\mathfrak{O}$, and then generalizing (3.2) to these ideals. With this in mind, begin by defining the special prime ideals of $\mathbb{Z}[\theta]$ that will be used in the algebraic factor base:

Definition 3.1.3. A *first degree* prime ideal $\mathfrak{p}$ of a Dedekind domain D is a prime ideal of D such that $\mathrm{N}(\mathfrak{p}) = p$ for some prime integer p.

Note 3.1.1. It should be observed that any ideal $\mathfrak{p}$ of a ring R with $\mathrm{N}(\mathfrak{p}) = p$ for some prime integer p is necessarily a prime ideal of R. This follows since $[R : \mathfrak{p}] = p$ implies that $R/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ is a field and hence $\mathfrak{p}$ is a maximal ideal of R [29, Lemma 5.1]. But any maximal ideal of R is also a prime ideal of R [29, Lemma 5.1].

Before proceeding to determine a good representation for the first degree prime ideals of $\mathbb{Z}[\theta]$, a technical lemma is in order:

Lemma 3.1.6. *If R is a commutative ring with identity 1_R, S is a commutative ring with identity 1_S, and $\phi : R \to S$ is a ring epimorphism, then $\phi(1_R) = 1_S$*

Proof. Let $y \in S$. Since ϕ is a ring epimorphism there exists $x \in R$ such that $\phi(x) = y$. Then $y \cdot \phi(1_R) = \phi(1_R) \cdot y = \phi(1_R) \cdot \phi(x) = \phi(1_R \cdot x) = \phi(x) = y$, hence $\phi(1_R) = 1_S$. □

The following result gives the convenient representation for the first degree prime ideals:

Theorem 3.1.7. *Let $f(x)$ be a monic, irreducible polynomial with integer coefficients and $\theta \in \mathbb{C}$ a root of $f(x)$. The set of pairs (r, p) where p is a prime integer and $r \in \mathbb{Z}/p\mathbb{Z}$ with $f(r) \equiv 0 \pmod{p}$ is in bijective correspondence with the set of all first degree prime ideals of $\mathbb{Z}[\theta]$.*

Proof. Let $\mathfrak{p}$ be a first degree prime ideal of $\mathbb{Z}[\theta]$. Then $[\mathbb{Z}[\theta] : \mathfrak{p}] = p$ for some prime integer p, so that $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$. There is a canonical epimorphism [10, Chapter 7, Theorem 7] of rings $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}[\theta]/\mathfrak{p}$ such that $\ker \phi = \mathfrak{p}$. Since $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ it follows that ϕ can also be thought of as an epimorphism of rings $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/p\mathbb{Z}$ with $\ker \phi = \mathfrak{p}$, that is, the elements in $\mathfrak{p}$ map to integers that are divisible by p, and any such integer is the image of an element in $\mathfrak{p}$. Furthermore $\phi(1) = 1$ by Lemma 3.1.6 and hence $\phi(a) \equiv a \pmod{p}$ for any integer a.

Let $r = \phi(\theta) \in \mathbb{Z}/p\mathbb{Z}$. If $f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + a_0$ with $a_i \in \mathbb{Z}$ for $0 \leq i < d$, then $\phi(f(\theta)) \equiv 0 \pmod{p}$ since $f(\theta) = 0$ and hence

$$\begin{aligned}
0 \equiv \phi(f(\theta)) &\equiv \phi(\theta^d + a_{d-1}\theta^{d-1} + \cdots + a_1\theta + a_0) \\
&\equiv \phi(\theta)^d + a_{d-1}\phi(\theta)^{d-1} + \cdots + a_1\phi(\theta) + a_0 \\
&\equiv r^d + a_{d-1}r^{d-1} + \cdots + a_1 r + a_0 \\
&\equiv f(r) \pmod{p}
\end{aligned}$$

so that r is a root of $f(x) \pmod{p}$ and the ideal $\mathfrak{p}$ determines the unique pair (r, p).

Conversely, let p be a prime integer and $r \in \mathbb{Z}/p\mathbb{Z}$ with $f(r) \equiv 0 \pmod{p}$. Then there is a natural ring epimorphism (analogous to the one discussed in Theorem 2.2.2) that maps polynomials in θ to polynomials in r. In particular, $\phi(a) \equiv a \pmod{p}$ for all $a \in \mathbb{Z}$ and $\phi(\theta) = r \pmod{p}$. Let $\mathfrak{p} = \ker \phi$ so that $\mathfrak{p}$ is an ideal of $\mathbb{Z}[\theta]$. Since ϕ is onto and $\ker \phi = \mathfrak{p}$ it follows that $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ and hence $[\mathbb{Z}[\theta] : \mathfrak{p}] = p$ and $\mathfrak{p}$ is therefore a first degree prime ideal of $\mathbb{Z}[\theta]$. Thus the pair (r, p) determines a unique first degree prime ideal $\mathfrak{p}$, which in turn determines the unique pair (r, p) consistent with the first part of this proof. This gives the result. □

The next step is to generalize (3.2) to prime ideals of $\mathbb{Z}[\theta]$ and to determine how this formula can be used in testing smoothness of an element $a + b\theta$ over an algebraic factor base. As it turns out, some of the properties of the exponents e_i in (3.2) can be generalized to exponents of first degree prime ideals of $\mathbb{Z}[\theta]$ occurring in the ideal factorization of $\langle \beta \rangle$ for $\beta \in \mathfrak{O}$. This is done by first observing that an exponent e_i in (3.2) can be thought of as a group homomorphism $e_{\mathfrak{p}_i} : \mathbb{Q}(\theta)^* \to \mathbb{Z}$, where $\mathbb{Q}(\theta)^*$ denotes the multiplicative group of non-zero elements in the field $\mathbb{Q}(\theta)$, for a fixed prime ideal $\mathfrak{p}_i$. This homomorphism has the following properties:

1. $e_{\mathfrak{p}_i}(\beta) \geq 0$ for all $\beta \in \mathbb{Q}(\theta)^*$.
2. $e_{\mathfrak{p}_i}(\beta) > 0$ if and only if the ideal $\mathfrak{p}_i$ divides the principal ideal $\langle \beta \rangle$.
3. $e_{\mathfrak{p}_i}(\beta) = 0$ for all but a finite number of prime ideals $\mathfrak{p}_i$ of $\mathfrak{O}$, and $|\,\mathrm{N}(\beta)| = \prod \mathrm{N}(\mathfrak{p}_i)^{e_{\mathfrak{p}_i}}$ for all prime ideals $\mathfrak{p}_i$ of D.

A non-trivial result [5, Proposition 5.4] using the Jordan-Hölder theorem establishes a group homomorphism possessing the properties outlined above, but defined for the prime ideals of $\mathbb{Z}[\theta]$ instead of the ideals of $\mathfrak{O}$:

Proposition 3.1.8. *For every prime ideal $\mathfrak{p}_i$ of $\mathbb{Z}[\theta]$, there exists a group homomorphism $l_{\mathfrak{p}_i} : \mathbb{Q}(\theta)^* \to \mathbb{Z}$ that possesses the following properties:*

1. *$l_{\mathfrak{p}_i}(\beta) \geq 0$ for all $\beta \in \mathbb{Q}(\theta)^*$.*
2. *$l_{\mathfrak{p}_i}(\beta) > 0$ if and only if the ideal $\mathfrak{p}_i$ divides the principal ideal $\langle \beta \rangle$.*
3. *$l_{\mathfrak{p}_i}(\beta) = 0$ for all but a finite number of prime ideals $\mathfrak{p}_i$ of $\mathbb{Z}[\theta]$, and $|\,\mathrm{N}(\beta)| = \prod \mathrm{N}(\mathfrak{p}_i)^{l_{\mathfrak{p}_i}}$ for all prime ideals $\mathfrak{p}_i$ of $\mathbb{Z}[\theta]$.*

In the GNFS, the only ideals $\mathbb{Z}[\theta]$ of concern are the principal ideals of the form $\langle a + b\theta \rangle$ for integers a and b, and because of this restriction the only homomorphisms of (3.1.8) that need to be considered are those corresponding to first degree prime ideals of $\mathbb{Z}[\theta]$, as the next result shows:

Theorem 3.1.9. *Given an element $\beta \in \mathbb{Z}[\theta]$ of the form $\beta = a + b\theta$ for coprime integers a and b and a prime ideal $\mathfrak{p}$ of $\mathbb{Z}[\theta]$, then the homomorphism $l_{\mathfrak{p}}$ of Proposition 3.1.8 corresponding to $\mathfrak{p}$ has $l_{\mathfrak{p}}(\beta) = 0$ if $\mathfrak{p}$ is not a first degree prime ideal of $\mathbb{Z}[\theta]$. Furthermore, if $\mathfrak{p}$ is a first degree prime ideal of $\mathbb{Z}[\theta]$ corresponding to the pair (r, p) as in Theorem 3.1.7, then*

$$l_{\mathfrak{p}}(\beta) = \begin{cases} \operatorname{ord}_p(\mathrm{N}(\beta)) & \text{if } a \equiv -br \pmod{p} \\ 0 & \text{otherwise} \end{cases}$$

where $\operatorname{ord}_p(\mathrm{N}(\beta))$ denotes the exponent of the prime integer p occurring in the unique factorization of the integer $\mathrm{N}(\beta)$ into distinct primes.

Proof. Let $\mathfrak{p}$ be a prime ideal of $\mathbb{Z}[\theta]$ with $l_{\mathfrak{p}}(a + b\theta) > 0$. Then $\mathfrak{p}$ serves as the kernel of the canonical epimorphism $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}[\theta]/\mathfrak{p}$. Now by Proposition 3.1.5 it follows that $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{F}_{p^e}$, where p is a prime integer, e is a positive integer, and $\mathbb{F}_{p^e}$ denotes the finite field with p^e elements. In particular, $\mathbb{Z}[\theta]/\mathfrak{p}$ must contain an isomorphic copy of the field $\mathbb{Z}/p\mathbb{Z}$. The strategy is to show $\operatorname{Im}\phi = \mathbb{Z}/p\mathbb{Z}$, for it then follows from $\mathbb{Z}[\theta]/\ker\phi \cong \operatorname{Im}\phi$ and $\ker\phi = \mathfrak{p}$ that $\mathbb{Z}[\theta]/\mathfrak{p} \cong \mathbb{Z}/p\mathbb{Z}$ and $\mathfrak{p}$ is a first degree prime ideal of $\mathbb{Z}[\theta]$.

Begin by noting that since ϕ is an epimorphism of rings, it follows from Lemma 3.1.6 that $\phi(1) = 1 \in \mathbb{Z}/p\mathbb{Z}$ and hence $\phi(m) \equiv m \pmod{p}$ for any integer m. Furthermore, note that the condition $l_{\mathfrak{p}}(a + b\theta) > 0$ implies that $\mathfrak{p}$ divides $\langle a + b\theta \rangle$ by Proposition 3.1.8 and hence $a + b\theta \in \mathfrak{p}$. But since $\ker\phi = \mathfrak{p}$ it follows that $\phi(a + b\theta) \equiv 0 \pmod{p}$.

Now suppose $b \in \mathbb{Z}$ is divisible by p. It then follows from $\phi(a + b\theta) \equiv 0 \pmod{p}$ and $\phi(b) \equiv b \equiv 0 \pmod{p}$ that

$$0 \equiv \phi(a + b\theta) \equiv a + b \cdot \phi(\theta) \equiv a \pmod{p} \tag{3.3}$$

and hence a is also divisible by p, contradictory to a and b being coprime. Thus b can't be divisible by p. Since b is not divisible by p it follows that b has a multiplicative inverse modulo p, denoted b^{-1}. Then (3.3) indicates that $a + b \cdot \phi(\theta) \equiv 0 \pmod{p}$ and hence $\phi(\theta) \equiv -ab^{-1} \pmod{p}$. The significance of the latter is that $\phi(\theta) \in \mathbb{Z}/p\mathbb{Z}$ and hence $\mathbb{Z}/p\mathbb{Z} \subseteq \phi(\mathbb{Z}[\theta]) \subseteq \mathbb{Z}/p\mathbb{Z}$ and thus $\operatorname{Im}\phi = \mathbb{Z}/p\mathbb{Z}$ and the first part of the result is proved.

To prove the second portion of this result, begin by proving that $l_{\mathfrak{p}}(a + b\theta) > 0$ for a first degree prime ideal $\mathfrak{p}$ with pair (r, p) if and only if $a \equiv -br \pmod{p}$. Suppose then that $l_{\mathfrak{p}}(a + b\theta) > 0$. By Proposition 3.1.8 it follows that $\mathfrak{p}$ divides $\langle a + b\theta \rangle$ and hence $a + b\theta \in \mathfrak{p}$. Now $\mathfrak{p} = \ker\phi$ where ϕ is the canonical epimorphism of Theorem 3.1.7 with $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/p\mathbb{Z}$ that sends $\phi(\theta) = r \pmod{p}$ and $\phi(a) \equiv a \pmod{p}$ for $a \in \mathbb{Z}$. Then $a + b\theta \in \mathfrak{p} = \ker\phi$ implies that $\phi(a + b\theta) \equiv 0 \pmod{p}$. But then $0 \equiv \phi(a + b\theta) \equiv a + br \pmod{p}$ and hence $a \equiv -br \pmod{p}$ as desired. Conversely, suppose $a \equiv -br \pmod{p}$ for the first degree prime ideal $\mathfrak{p}$ with pair (r, p). Then $0 \equiv a + br \pmod{p} \equiv \phi(a + b\theta)$ and hence $a + b\theta \in \ker\phi = \mathfrak{p}$. But the latter implies that $\mathfrak{p}$ divides $\langle a + b\theta \rangle$ and hence $l_{\mathfrak{p}}(a + b\theta) > 0$ by Proposition 3.1.8.

It should be noted that the result of the proceeding paragraph can also be stated such that if $\mathfrak{p}$ is a first degree prime ideal of $\mathbb{Z}[\theta]$ with pair (r,p), then $l_{\mathfrak{p}}(a+b\theta)=0$ if and only if $a \not\equiv -br \pmod{p}$.

Next, it will be shown that for a first degree prime ideal $\mathfrak{p}$ of $\mathbb{Z}[\theta]$ with pair (r,p) that $\mathrm{N}(a+b\theta)$ is divisible by p if and only if $a \equiv -br \pmod{p}$. Combining this with earlier work yields $l_{\mathfrak{p}}(a+b\theta)>0$ if and only if p divides $\mathrm{N}(a+b\theta)$, which justifies using the norm as a smoothness test for an algebraic factor base of first degree prime ideals.

Recalling the definition of the norm from (3.1) and the embeddings of Theorem 3.1.2 that comprise it, the following explicit computation of the norm for an element of the form $a+b\theta$ sheds light on when a prime p divides $\mathrm{N}(a+b\theta)$:

$$\begin{aligned}\mathrm{N}(a+b\theta) &= \sigma_1(a+b\theta)\cdot\sigma_2(a+b\theta)\cdots\sigma_d(a+b\theta)\\ &= (a+b\theta_1)\cdot(a+b\theta_2)\cdots(a+b\theta_d)\\ &= b^d\left[\left(\frac{a}{b}+\theta_1\right)\cdot\left(\frac{a}{b}+\theta_2\right)\cdots\left(\frac{a}{b}+\theta_d\right)\right]\\ &= (-b)^d\left[\left(-\frac{a}{b}-\theta_1\right)\cdot\left(-\frac{a}{b}-\theta_2\right)\cdots\left(-\frac{a}{b}-\theta_d\right)\right]\\ &= (-b)^d f\left(-\frac{a}{b}\right)\end{aligned} \tag{3.4}$$

From (3.4) it follows that a prime p divides $\mathrm{N}(a+b\theta)$ if and only if p divides either $(-b)^d$ or $f(-a/b)$. But since p does not divide b it follows that $f(-a/b)\equiv 0 \pmod{p}$ and hence $a\equiv -br \pmod{p}$ for some root r of $f(x) \pmod{p}$. The value for r, taken together with p, determines a first degree prime ideal pair for which $l_{\mathfrak{p}}(a+b\theta)>0$, and vice versa.

To complete the result, suppose $l_{\mathfrak{p}}(a+b\theta)>0$ for some first degree prime ideal $\mathfrak{p}$ of $\mathbb{Z}[\theta]$ with pair (r,p). Further suppose that another first degree prime ideal $\mathfrak{p}_2$ exists with pair (r_2,p) such that $l_{\mathfrak{p}_2}(a+b\theta)>0$. Then it follows that $a\equiv -br \pmod{p}$ and $a\equiv -br_2 \pmod{p}$. But the latter implies that $r\equiv r_2 \pmod{p}$ and hence $\mathfrak{p}$ and $\mathfrak{p}_2$ correspond to the same pair and hence represent the same ideal. Thus, for any prime p there can be at most one first degree prime ideal $\mathfrak{p}$ which has p in its pair (r,p) and that has $l_{\mathfrak{p}}(a+b\theta)>0$ for fixed a and b. In particular, if such an ideal exists, it must account for all the powers of p in $\mathrm{N}(a+b\theta)$ by Proposition 3.1.8 and hence $l_{\mathfrak{p}}(a+b\theta)=\mathrm{ord}_p(\mathrm{N}(a+b\theta))$. □

Theorem 3.1.9 is important for two reasons. First, it proves that the only prime ideals of $\mathbb{Z}[\theta]$ occurring in the ideal factorization of a principal ideal of the form $\langle a+b\theta\rangle$ for coprime integers a and b are the first degree prime ideals of $\mathbb{Z}[\theta]$. Secondly, and probably even more important, this result gives a condition for determining whether a first degree prime ideal occurs in the ideal factorization of $\langle a+b\theta\rangle$. Specifically, a first degree prime ideal corresponding to the pair (r,p) as in Theorem 3.1.7 occurs as a non-trivial factor in the ideal factorization of $\langle a+b\theta\rangle$ if and only if $a\equiv -br \pmod{p}$. This is a fairly easy condition to check, and indeed gives rise to sieving a sieving procedure outlined in §3.7. To summarize, finding an element $a+b\theta\in\mathbb{Z}[\theta]$ that is smooth over an algebraic factor base of first degree prime ideals of $\mathbb{Z}[\theta]$ amounts to finding an element $a + b\theta$ such that the integer $\mathrm{N}(a + b\theta)$ factors completely over the primes occurring in the (r, p) pairs corresponding to the first degree prime ideals in the algebraic factor base.

To begin to see how Theorem 3.1.9 can be used to produce a square in $\mathbb{Q}(\theta)$, and hence a square in $\mathfrak{O}$ by [5, pages 60–61]), note the following result:

Theorem 3.1.10. *If U is a set of pairs of integers (a, b) such that the product of all elements $a + b\theta \in \mathbb{Z}[\theta]$ is a perfect square $\alpha^2 \in \mathbb{Q}(\theta)$, then*

$$\sum_{(a,b)\in U} l_{\mathfrak{p}_i}(a + b\theta) \equiv 0 \pmod{2} \tag{3.5}$$

for all prime ideals $\mathfrak{p}_i$ of $\mathbb{Z}[\theta]$.

Proof. Let $\mathfrak{p}_i$ be any prime ideal of $\mathbb{Z}[\theta]$. By Proposition 3.1.8 $l_{\mathfrak{p}_i}$ is a homomorphism from the multiplicative group of nonzero elements $\mathbb{Q}(\theta)^*$ to the additive group of integers $\mathbb{Z}$and hence:

$$\sum_{(a,b)\in U} l_{\mathfrak{p}_i}(a + b\theta) = l_{\mathfrak{p}_i}\left(\prod_{(a,b)\in U}(a + b\theta)\right) = l_{\mathfrak{p}_i}\left(\alpha^2\right) = 2l_{\mathfrak{p}_i}(\alpha) \equiv 0 \pmod{2} \quad \square$$

Note that this result gives a necessary condition for a product of elements of the form $a + b\theta$ to be a square in $\mathbb{Q}(\theta)$, but not a sufficient one. More explicitly, when a number of (a, b) pairs with $a + b\theta$ smooth over an algebraic factor base is found exceeding the number of ideals in the algebraic factor base, linear algebra may be applied as outlined in §1.3 to produce a subset U of (a, b) pairs such that

$$\sum_{(a,b)\in U} l_{\mathfrak{p}_i}(a + b\theta) \equiv 0 \pmod{2}$$

for all ideals $\mathfrak{p}_i$ of $\mathbb{Z}[\theta]$. This does not necessarily mean that the product of the elements $a + b\theta$ in U is a square in $\mathbb{Z}[\theta]$, though. This condition *can* be made sufficient with a little bit of extra work, as will be seen in §3.2

3.2 Quadratic Characters

When a set U of pairs of integers (a, b) has been found such that (3.5) holds, a further test is needed to determine whether or not the product of the corresponding elements $a + b\theta \in \mathbb{Z}[\theta]$ is a perfect square in $\mathbb{Z}[\theta]$. This problem is solved in a straight-forward and efficient manner [1, 5] through the use of "quadratic characters." To motivate the discussion of quadratic characters, a simple scheme for squareness testing in $\mathbb{Z}$ is illustrated.

Begin by noting that if x is an integer in $\mathbb{Z}$ such that $x = y^2$ for some integer y, then x is also a perfect square modulo p for every prime p. This is the case since for any odd prime p

$$\left(\frac{x}{p}\right) \equiv x^{\frac{p-1}{2}} \equiv y^{\frac{2(p-1)}{2}} \equiv y^{p-1} \equiv 1 \pmod{p}$$

by Euler's Criterion [24, Corollary 2.38] and the fact that the non-zero elements of $\mathbb{Z}/p\mathbb{Z}$ form a multiplicative group of order $p-1$ [14, Chapter 5, Theorem 5.3]. Note that any integer is a perfect square modulo 2.

Thus, given any finite set of primes it follows that if an integer x is a perfect square then it is also a perfect square modulo those primes. Although the converse is not true, one idea that may work well on very lose probabilistic grounds is that if an integer x is a perfect square modulo a number of primes p, then x itself is a perfect square. The certainty of the method can be increased, in some sense, by increasing the number of primes that x is tested against.

To illustrate with the integer 79 and the set $\{2, 3, 5, 7\}$ of primes, it is easily seen that $79 \equiv 1^2 \pmod{2}$, $79 \equiv 1^2 \pmod{3}$, $79 \equiv 2^2 \pmod{5}$, and $79 \equiv 4^2 \pmod{7}$. Thus, 79 is a perfect square modulo all the primes in the aforementioned set, yet is not a perfect square itself. If 11 is added to the set of test primes, it is seen that $79^5 \equiv -1 \pmod{11}$ and hence 79 is not a perfect square modulo 11 by Euler's Criterion and therefore 79 is not a perfect square in $\mathbb{Z}$. Thus, one more prime in the set of test primes in this example would have prevented the misidentification of 79 as a perfect square when using the smaller test set.

The following result generalizes this test for perfect squares in $\mathbb{Z}$ to $\mathbb{Q}(\theta)$:

Theorem 3.2.1. *Let U be a set of (a, b) pairs such that*

$$\prod_{(a,b)\in U} (a + b\theta) = \alpha^2$$

for some $\alpha \in \mathbb{Q}(\theta)$. Given a first degree prime ideal $\mathfrak{q}$ corresponding to the pair (s, q) that does not divide $\langle a + b\theta \rangle$ for any pair (a, b) and for which $f'(s) \not\equiv 0 \pmod{q}$, it follows that

$$\prod_{(a,b)\in U} \left(\frac{a + bs}{q}\right) = 1 \tag{3.6}$$

Proof. Let $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/q\mathbb{Z}$ with $\phi(\theta) = s \pmod{q}$ be the canonical ring epimorphism of Theorem 3.1.7. Then $\mathfrak{q} = \ker \phi$ is the first degree prime ideal corresponding to the pair (s, q). Note that restricting ϕ to the members of $\mathbb{Z}[\theta]$ which are not in $\mathfrak{q}$ maps onto the non-zero elements of $\mathbb{Z}/q\mathbb{Z}$, which allows for the definition of the map $\chi_{\mathfrak{q}} : \mathbb{Z}[\theta] - \mathfrak{q} \to \{1, -1\}$ given by

$$\chi_{\mathfrak{q}}(\gamma) = \left(\frac{\phi(\gamma)}{q}\right).$$

From the remarks made in Note 2.4.1 it follows that exists a $\beta = f'(\theta) \cdot \alpha \in \mathbb{Z}[\theta]$ satisfies

$$f'(\theta)^2 \cdot \prod_{(a,b)\in U} (a + b\theta) = \beta^2$$

By the hypothesis that $\langle a + b\theta \rangle$ is not divisible by the ideal $\mathfrak{q}$ it follows that $a + b\theta \notin \mathfrak{q}$. Similarly, since $f'(s)$ is assumed to not be divisible by q then $f'(\theta)^2 \notin \mathfrak{q}$. Thus $\langle \beta^2 \rangle$ is not divisible by $\mathfrak{q}$ and neither is $\langle \beta \rangle$ and hence $\chi_{\mathfrak{q}}$ is defined at β^2 and β.

Using the elementary properties of the Legendre symbol it is seen that

$$\chi_{\mathfrak{q}}(\beta^2) = \left(\frac{\phi(\beta^2)}{q}\right) = \left(\frac{\phi(\beta) \cdot \phi(\beta)}{q}\right) = \left(\frac{\phi(\beta)}{q}\right)^2 = 1$$

and similarly $\chi_{\mathfrak{q}}(f'(\theta)^2) = 1$. But then

$$\begin{aligned}
1 = \chi_{\mathfrak{q}}(\beta^2) = \chi_{\mathfrak{q}}\left(f'(\theta)^2 \cdot \prod_{(a,b)\in U} (a + b\theta)\right) &= \left(\frac{\phi\left(f'(\theta)^2 \cdot \prod_{(a,b)\in U}(a + b\theta)\right)}{q}\right) \\
&= \left(\frac{\phi\left(f'(\theta)^2\right) \cdot \prod_{(a,b)\in U} \phi(a + b\theta)}{q}\right) = \left(\frac{\phi\left(f'(\theta)^2\right)}{q}\right) \cdot \left(\frac{\prod_{(a,b)\in U} \phi(a + b\theta)}{q}\right) \\
&= \chi_{\mathfrak{q}}\left(f'(\theta)^2\right) \cdot \left(\frac{\prod_{(a,b)\in U}(a + bs)}{q}\right) = 1 \cdot \prod_{(a,b)\in U} \left(\frac{a + bs}{q}\right)
\end{aligned}$$

and the result follows. □

Just like Theorem 3.1.10, this result gives a necessary condition for squareness in $\mathbb{Q}(\theta)$ but not a sufficient one. But given a set U of (a, b) pairs and a set Q of first degree prime ideals of $\mathbb{Z}[\theta]$ that satisfy both (3.5) and Theorem 3.6, it follows [5, page 70] that the product of all the elements $a + b\theta$ corresponding to (a, b) pairs in U is very *likely* to be a perfect square in $\mathbb{Q}(\theta)$. As in the analogous test for square in $\mathbb{Z}$, increasing the number of ideals in Q also increases the likelihood of identifying squares correctly.

In further discussions a set Q of first degree prime ideals of $\mathbb{Z}[\theta]$ satisfying the hypothesis of Theorem 3.2.1 is referred to as a *quadratic character base*, and the corresponding maps $\chi_{\mathfrak{q}}$ are called *quadratic characters.*

3.3 Summary of Finding Squares in $\mathbb{Z}[\theta]$

To pull together the material that has been developed so far, a primary goal of the GNFS is to find a set U of pairs of integers (a, b) such that (2.1) holds. This is done by first selecting an algebraic factor base I consisting of a finite number of first degree prime ideals of $\mathbb{Z}[\theta]$. A quadratic character base Q of first degree prime ideals whose corresponding (s, q) pairs satisfy the hypothesis of Theorem 3.2.1 is also chosen.

Next, (a, b) pairs are found for which the principal ideals $\langle a+b\theta\rangle$ factor completely into ideals in I, using a sieving procedure detailed in §3.7. When the number of (a, b) pairs exceeds the number of ideals in the algebraic factor base and the quadratic character base, linear algebra may be used to find a subset U of those pairs that satisfy (3.5) for all $\mathfrak{p}_i \in I$ and (3.6) for all $\mathfrak{q} \in Q$. These latter two conditions ensure that (2.1) holds and hence a square in $\mathbb{Z}[\theta]$ can be found by Note 2.4.1.

3.4 The Rational Factor Base and Sieving

Besides finding a perfect square in $\mathbb{Z}[\theta]$, the GNFS algorithm simultaneously requires a square in $\mathbb{Z}$ be found. Just as a sieving procedure was naturally constructed in §1.3 because of the special form of the polynomial $f(r_i) = r_i^2 - n$, a sieve can be used to find pairs of integers (a, b) with $a + bm$ smooth over a "rational" factor base F because of the special form of $a + bm$. Note the term "rational" is applied to the factor base F to distinguish it from the algebraic factor base I of first degree prime ideals of $\mathbb{Z}[\theta]$ defined in §3.1.

The first obstacle to get around involves the fact that there are two free variables a and b that can be adjusted when looking for a smooth $a + bm$ whereas in the QS there was just the single r_i that was variable in $f(r_i) = r_i^2 - n$. Most implementations of the GNFS simply fix a value for b and then scan the a values within a range $u < a < u$ for values of $a + bm$ that are smooth, just as values of r_i are scanned within some range $u < r_i < u$ in the QS. Note that the value of b usually starts at 1 and is incremented until enough (a, b) pairs have been found with $a + bm$ smooth in order to facilitate the linear algebra step for producing squares in $\mathbb{Z}[\theta]$ and $\mathbb{Z}$.

To see how a sieving procedure can be used, let p be a fixed prime in the rational factor base F and b a fixed, positive integer. Then for any $a \in \mathbb{Z}$ the prime p divides $a + bm$ if and only if $a + bm \equiv 0 \pmod{p}$. This implies that $a \equiv -bm \pmod{p}$ and hence a must be of the form $a = -bm + kp$ for some $k \in \mathbb{Z}$. This observation gives a clean representation for the possible (a, b) pairs that have $a + bm$ divisible by p for a fixed prime p and positive integer b.

Sieving over F in the GNFS then begins with a "sieve array" of computer memory with a single position allocated for each $-u < a < u$. For a fixed value of b, each position in the sieve array is initialized with the appropriate value of $a+bm$ for that position. For each prime $p \in F$, one computes the finite set of values $a = -bm + kp$ for $k \in \mathbb{Z}$ with $-u < a < u$, and for each such a divides the prime p out of the number stored in the position corresponding to a in the sieve array. After this has been performed for all primes in F, the sieve array is scanned for values of 1. Any such position in the sieve array yields a value for a for which

$a + bm$ is smooth over F. This procedure is then continued for the next value of b until enough pairs (a, b) have been found with $a + bm$ smooth to allow for the linear algebra step.

3.5 Speeding Up The Sieve

The sieve outlined in §3.4 is a reasonably fast procedure because it greatly reduces the number of divisions that must be performed. Specifically, instead of blindly dividing values of $a + bm$ by every prime in F, this sieving procedure will only divide by a prime p when it is guaranteed that $a + bm$ is divisible by p. It still often happens that $a + bm$ is not smooth over F, though, and when that is the case the divisions represent wasted time. A clever rearrangement of the sieving procedure keeps the number of time-consuming divisions to a minimum by effectively replacing the common division operations with faster additions. Instead of using division to facilitate the smoothness test over F, additions will be used to rule out most $a + bm$ values that are not smooth, and trial division will be performed only on values which are almost certain to be smooth. Note that trial division is still used in order to guarantee that an $a + bm$ really is smooth over F.

This basic idea leads to storing approximations to $\ln(a+bm)$ in the sieve array instead of the actual value $a+bm$. From the elementary theory of logarithms, dividing $a+bm$ by a prime p is equivalent to subtracting $\ln(p)$ from $\ln(a + bm)$. Thus, for a fixed b and prime p, one subtracts $\ln(p)$ from the array location for $a = -bm + kp$ for $k \in \mathbb{Z}$ with $-u < a < u$. After processing all primes in F for a fixed b, the sieve array is then scanned for values $\leq 0 = \ln(1)$ instead of 1. Such a position yields a value for a with the value $a + bm$ very "likely" to be smooth. Smoothness is then tested on such $a+bm$ by performing trial division over F. The term "likely" is used because in some cases, due to the round-off errors in approximating logarithms, some $a + bm$ values will turn out to not be smooth over F. These occurrences are infrequent in practice, and in any event are negligible compared to the savings in time brought about by not performing divisions on a large number of $a + bm$ values.

3.6 Implementation Techniques For Speeding Up The Sieve

When actually implementing the sieving technique described in §3.5, there are a few improvements which lead to performance gain in practice, without dramatically altering the basic method.

The most common adjustment made is to initialize the sieve array with $-\ln(a+bm)$ instead of $\ln(a + bm)$, and the values of $\ln(p)$ are *added* to the sieve array positions instead of being subtracted. When scanning the sieve array for $a + bm$ values that are probably smooth, one can then perform a non-negativity test on the sieve array positions to determine the values that will be further tested with trial division. On many architectures this operation is faster than determining if a sieve array position is less than or equal to zero as is required in §3.5. Furthermore, it is usually the case that the approximations to $\ln(a + bm)$ can fit within a single byte (8 bits) of computer memory, and so the non-negativity test can be done four positions at a time if the architecture has 32 bit words, 8 positions at a time if there are 64 bit words, and so on. This can lead to a slight performance gain.

Another practical improvement comes from the fact that in most cases where GNFS is applied, the integer m is significantly larger than a and b. As such, m is the dominant component of $\ln(a + bm)$, so instead of computing $\ln(a + bm)$ for every (a, b) pair, one can simply compute $\ln(bm)$ for a fixed b and initialize the sieve array to $-\ln(bm)$. This saves the time of computing a large number of logarithms, but should be used with discretion since it further adds to the round-off error already present in the approximations to the logarithms. The consequences of the latter are that some smooth $a + bm$ values may be missed, and conversely more non-smooth $a + bm$ values may be trial-divided than would ordinarily be the case.

While on the topic of the errors in using approximations $-\ln(bm)$, another inaccuracy is introduced by using logarithms when $a + bm$ values exist that are divisible by powers of primes p in F. Specifically, if $a + bm$ is divisible by p^e with $p \in F$ and $e > 1$, then $e \cdot \ln(p)$ should be added to the sieve array position for that (a, b) pair, not just $\ln(p)$. Not doing this can cause the sieve array position for this (a, b) pair to "come up short" and be deemed unlikely to be smooth, even if it actually is. Thus $a + bm$ values which are not square-free could be missed by this procedure.

In an attempt to account for these inaccuracies that arise when using logarithms and not sieving with prime powers, most implementations of GNFS initialize the sieve array with $-\ln(bm) + B$ instead of $-\ln(bm)$ for some "fudge factor" B. The purpose of the constant B is to decrease both the number of smooth $a + bm$ values that are missed and the number of non-smooth $a + bm$ values that are trial divided. Selecting a good value for B is very implementation and situation specific and hence requires some degree of experimentation at the initial stages of a factorization attempt with GNFS.

3.7 Sieving with the Algebraic Factor Base

Sieving with the algebraic factor base I proceeds in exactly the same manner as outlined in §3.4 because of the convenient representation of first degree prime ideals of $\mathbb{Z}[\theta]$ as pairs of integers (r, p) according to Theorem 3.1.7. Recall from the proof of Theorem 3.1.9 that a first degree prime ideal $\mathfrak{p}$ represented by the pair (r, p) divides $\langle a + b\theta \rangle$ if and only if p divides $\mathrm{N}(a + b\theta)$, which occurs if and only if $a \equiv -br \pmod{p}$. Thus, for a fixed b and first degree prime ideal $\mathfrak{p} \in I$ represented by the pair (r, p), it follows that the (a, b) pairs with $\langle a + b\theta \rangle$ divisible by $\mathfrak{p}$ must have a of the form $a = -br + kp$ for some $k \in \mathbb{Z}$.

These observations lead to the same sieving procedure in §3.4, with some minor modifications. First, each sieve array position is initialized with the value for $\mathrm{N}(a + b\theta)$ instead of $a + bm$, since the norm is used to test for smoothness of $a + b\theta$. Secondly, when logarithms are used as in §3.5, no single initializer can be used like $\ln(-bm)$ was because of the high degree of variability [19, pages 26–28] of $\mathrm{N}(a+b\theta)$ for different values of a. The immediate alternative is computing $\ln(\mathrm{N}(a+b\theta))$ for each sieve array position, which can waste a great deal of time on (a, b) pairs for which $a+b\theta$ is not smooth. By using a "fudge factor" similar to B in §3.6, though, one can avoid explicitly computing $\ln(\mathrm{N}(a + b\theta))$ when $a + b\theta$ is not smooth [19, pages 26–28].

3.8 An Implementation Note

As a practical matter, since the elements of the algebraic factor base and quadratic character base can be stored as integer pairs by Theorem 3.1.7, the rational factor base can be stored in a similar manner as pairs $(m \pmod p), p)$. This is possible since for a fixed b, the values for of a for which $a+bm$ is divisible by a prime p are of the form $a = -bm+kp$ for $k \in \mathbb{Z}$ from §3.4, just as the values of a for which $\langle a+b\theta \rangle$ is divisible by the first degree prime ideal represented by the pair (r, p) are of the form $a = -br + kp$ for $k \in \mathbb{Z}$ from §3.7. The importance of this is that the same basic sieving code in an implementation of the GNFS can be used both with the rational factor base and the algebraic factor base since the representation and treatment of the elements are the same.

Chapter 4

Filling in the Details

In this section, we will attempt to explain some of the issues not addressed by the GNFS algorithm detailed in Chapter 3. This includes finding suitable choices for the degree d of the polynomial $f(x)$, the polynomial $f(x)$ itself, and an integer m with $f(m) \equiv 0 \pmod{n}$, all addressed in §4.1. Computing the algebraic factor base and the quadratic character base is discussed in §4.2. The linear algebra step is explained in §4.3, §4.4, and §4.5.

One of the more difficult problems not addressed by the basic GNFS algorithm is computing the value of $\phi(\beta) = x \in \mathbb{Z}/n\mathbb{Z}$ for the $\beta \in \mathbb{Z}[\theta]$ of Note 2.4.1, given that only the value $\delta = \beta^2 \in \mathbb{Z}[\theta]$ is initially known. This at first seems like a straightforward task since δ may be considered as a polynomial $\delta(\theta)$ in θ of degree less than d, and similarly for β. Computing β is then a matter of applying any one of a number of techniques for factoring polynomials, specifically factoring the polynomial $x^2 - \delta(x)$. Recalling that the natural homomorphism $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/n\mathbb{Z}$ is defined by $\phi(\theta) = m \pmod{n}$, it follows that x can be found by substituting m for θ in $\beta(\theta)$ and reducing modulo n.

The difficulty that prevents any of the standard, straightforward approaches from being feasible is the size of the coefficients of the polynomial $\delta(\theta)$. Specifically, δ is computed as the product of the $a + b\theta$ values corresponding to the (a, b) pairs found in the linear algebra step of the algorithm. In the cases where the GNFS algorithm is applied, factor bases with hundreds of thousands of elements are used, with the consequence being that δ could be the product of tens of thousands of $a + b\theta$ values. This leads to obvious coefficient explosion of $\delta(\theta)$ and makes doing even the simplest arithmetical operations on δ intractable.

One way around the problem of computing β in $\mathbb{Z}[\theta]$ is to work in related fields where the computations *are* feasible. As will be seen in §4.6, finite fields $\mathbb{F}_{p^d}$ with p^d elements corresponding to $\mathbb{Q}(\theta)$ can be introduced and β computed in these restricted domains. Through a clever use of the Chinese Remainder Theorem, the resulting $\phi(\beta) = x \in \mathbb{Z}/n\mathbb{Z}$ can be computed easily and efficiently.

This square root method does introduce two subproblems of its own, namely finding applicable finite fields for $\mathbb{Q}(\theta)$ and computing square roots in those fields. In §4.6 it will be seen that finding the required finite fields amounts to testing the polynomial $f(x)$ for irreducibility modulo p for various primes p, which is a well-known problem whose solution is addressed in §4.9. Finding square roots in these finite fields also turns out to be easily accomplished through an adaptation of the Shanks method [16, Section 9.2] for finding square roots of integers modulo primes, explained in §4.8.

4.1 Finding a Polynomial

The basic GNFS algorithm outlined in Chapter 3 requires a monic, irreducible polynomial $f(x)$ of degree d with integer coefficients and which has a root m modulo n, where n denotes the integer to factor. However, no method is given for finding the optimal degree d, the polynomial $f(x)$, or the root m. The algorithm itself functions the same regardless of the the selections for these parameters, so it does make sense to leave methods for making these choices unspecified. In practice a number of techniques are used in the search for "good" initial parameters, because with careful experimentation and parameter adjustment, the time required to factor an integer n can be dramatically reduced. Finding choices for d, m and the polynomial $f(x)$ that lead to many (a, b) pairs for which $a + bm$ and $a + b\theta$ are smooth is very much an underdeveloped subject, and is currently one of the most active areas in GNFS research.

In most implementations of the GNFS, parameter selection begins with a choice for d. Experimentation and experience [8] have dictated that for factoring an integer with more than 110 digits, the degree d be set to 5. For integers between 50 and 80 digits a value of 3 for d is used. A degree value of 4 has faired well for integers with between 80 and 110 digits, but for reasons discussed in §4.6, early implementations of GNFS restricted d to an odd integer. In this case, $d = 5$ is usually substituted for $d = 4$.

Having selected a value for d, the choice of $f(x)$ and m is usually made simultaneously. First m is chosen with $m \approx n^{1/d}$ and such that the quotient of n divided by m^d is exactly one. A "base-m" expansion [5, Section 3] of n then gives

$$n = m^d + a_{d-1}m^{d-1} + \cdots + a_1 m + a_0$$

with coefficients $0 \leq a_i < m$ for $0 \leq i < d$. These coefficients may then be used to construct

$$f(x) = x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + a_0$$

which is monic of degree d. By construction $f(m) = n \equiv 0 \pmod{n}$ so that m is a root modulo n of $f(x)$. Furthermore, if $f(x)$ is reducible then $f(x) = g(x) \cdot h(x)$ for non-constant polynomials $g(x)$ and $h(x)$ and it follows that

$$n = f(m) = g(m) \cdot h(m)$$

is likely [5, page 54] to yield a non-trivial factorization of n. Thus, if $f(x)$ is reducible then n is likely is to be factored and the whole procedure can terminate, or $f(x)$ is irreducible and the sieving procedures of the GNFS can commence.

As will be seen in Chapter 6, this basic procedure can be expanded upon to give a range of different values for m and $f(x)$ to experiment with. In practice, because the high degree of variability of smoothness associated with (a, b) pairs for different polynomials, it is beneficial to experiment with different candidate values for $f(x)$ and m before committing to particular selection of parameters.

4.2 Finding First Degree Prime Ideals of $\mathbb{Z}[\theta]$

Finding first degree prime ideals of $\mathbb{Z}[\theta]$ for the algebraic factor base I and the quadratic character base Q amounts to finding integer pairs (r, p) with p a prime and r satisfying $f(r) \equiv 0 \pmod{p}$ according to Theorem 3.1.7. In other words, finding first degree prime ideals is equivalent to finding roots of $f(x)$ modulo p for various prime integers p. Fortunately, this happens to be a well-studied problem which can be solved in a natural and efficient way [21, Chapter 4, Section 3].

A naive approach to finding roots of the polynomial $f(x) \pmod{p}$ is to simply "plug in" all the integers from 0 to $p-1$ and determine which values are mapped to 0 by $f(x)$. As with most brute-force approaches, this works well for a small number of cases, specifically when p is "small", but becomes quite impractical for the larger values of p used in the GNFS.

A dramatic improvement over this brute-force method can be made using the following result in a clever way:

Theorem 4.2.1. *When considered as a polynomial in $\mathbb{Z}/p\mathbb{Z}[x]$, the polynomial $x^p - x$ factors as*

$$x^p - x = \prod_{i=0}^{p-1}(x - i) \tag{4.1}$$

Proof. It's an elementary result [14, Chapter 5, Theorem 5.3] from abstract algebra that the non-zero elements of a field form a group under multiplication. In this case, that means the $p-1$ non-zero elements of $\mathbb{Z}/p\mathbb{Z}$ form a finite group of order $p-1$ under multiplication. Then for any $0 < a < p$ it follows that $a^{p-1} \equiv 1 \pmod{p}$ and therefore $a^p \equiv a \pmod{p}$ for all a with $0 \leq a < p$. Rearranging the last congruence yields $a^p - a \equiv 0 \pmod{p}$ and therefore a is seen to be a root of $x^p - x \pmod{p}$ for $0 \leq a < p$. This determines p roots for $x^p - x \pmod{p}$. But $x^p - x \pmod{p}$ has at most p roots and hence has exactly the roots enumerated. Each root of $x^p - x$ determines a monic, linear factor of $x^p - x \pmod{p}$ and vice versa so (4.1) follows. □

Since finding roots of $f(x) \pmod p$ is equivalent to finding monic, linear factors of $f(x) \pmod p$, and $x^p - x \pmod p$ is the product of all the monic, linear polynomials over $\mathbb{Z}/p\mathbb{Z}$ by (4.1), a natural idea is to somehow use $x^p - x \pmod p$ in the root finding procedure. With this in mind, the first realization is that finding roots of $f(x) \pmod p$ is equivalent to finding roots of $g(x) = \gcd(f(x), x^p - x)$. The effect of computing $g(x) \pmod p$ is to isolate the portion of $f(x) \pmod p$ which is the product of monic, linear polynomials over $\mathbb{Z}/p\mathbb{Z}$, since this is the portion where the roots of $f(x) \pmod p$ are to be found. Another way of thinking of this computation is as a way to "strip out" of $f(x) \pmod p$ any quadratic or higher degree polynomials that occur in its canonical factorization into irreducibles, since such polynomials have nothing to do with the roots of $f(x) \pmod p$.

Now let b be any random integer with $0 \leq b < p$. Since $g(x) \pmod p$ divides $x^p - x \pmod p$ it must be a product of distinct, monic, linear polynomials, and therefore so is $g(x-b) \pmod p$. If x is a factor of $g(x-b) \pmod p$ then $g(-b) \equiv 0 \pmod p$ so a root $-b$ of $g(x) \pmod p$ and hence of $f(x) \pmod p$ has been found. On the other hand, if x is not a factor of $g(x-b) \pmod p$ then

$$g(x-b) \,|\, x^p - x = x(x^{p-1} - 1) = x\big(x^{(p-1)/2} + 1\big)\big(x^{(p-1)/2} - 1\big)$$

and the factors of $g(x-b) \pmod p$ fall between $(x^{(p-1)/2} + 1) \pmod p$ and $(x^{(p-1)/2} - 1) \pmod p$. If not all of the factors of $g(x-b) \pmod p$ divide into either of these latter polynomials, i.e. if $x^{(p-1)/2} \not\equiv \pm 1 \pmod{g(x-b)}$, then $g(x-b) \pmod p$ can be split non-trivially into the polynomials $g_1(x) = \gcd(g(x-b), x^{(p-1)/2} + 1)$ and $g_2(x) = \gcd(g(x-b), x^{(p-1)/2} - 1)$, with the degree of each polynomial strictly less than the degree of $g(x) \pmod p$.

If $g_i(x) \pmod p$ is a monic polynomial then a root of $g(x) \pmod p$ has been found. Otherwise, the same procedure outlined above is applied to each $g_i(x) \pmod p$ to split them into lesser degree polynomials. The algorithm continues on in this manner until it terminates, having found all the roots of $f(x) \pmod p$. This procedure is guaranteed to terminate since polynomials are produced at each stage with degrees strictly less than the degrees of the polynomials of the previous stage.

Note 4.2.1. In the event that $x^{(p-1)/2} \equiv \pm 1 \pmod{g(x-b)}$, other values for b are substituted until this condition no longer holds. Also note that a root r of $g(x-b) \pmod p$ gives rise to the root $r - b$ of $g(x) \pmod p$, and that r itself is not a root of $g(x) \pmod p$ unless $b = 0$.

4.3 Matrices and Dependencies

In the QS detailed in §1.3, each value of $f(r_i) = r_i^2 - n$ that is smooth over the factor base F is equated with a binary vector determined by the parity of the exponents of the primes $p \in F$ occurring in the factorization of $f(r_i) = r_i^2 - n$. Binary vectors $e_{(a,b)}$ for the (a, b) pairs occurring in the GNFS for which $a + bm$ and $a + b\theta$ are smooth over the rational and algebraic factor bases, respectively, are determined [5, pages 69–70] from the factorizations of $a+bm$ into prime integers and $a+b\theta$ into first degree prime ideals of $\mathbb{Z}[\theta]$. Each binary vector $e_{(a,b)}$ is also augmented with information relating a particular $a+b\theta$ to the quadratic character base, detailed in §3.2, that will ensure a product of $a+b\theta$ values that is a square in $\mathfrak{O}$. If there are k primes in the rational factor base, l first degree prime ideals of $\mathbb{Z}[\theta]$ in the algebraic factor base, and m first degree prime ideals in the quadratic character base, then each $e_{(a,b)}$ will be comprised of $1+k+l+m$ binary bits, determined by the sign of $a+bm$ and the respective bases. When these binary vectors are grouped together as columns in a matrix B, the binary vector resulting from the addition of the two columns for pairs (a,b) and (c,d) represents $(a+bm)\cdot(c+dm)$ and $\langle a+b\theta\rangle\cdot\langle c+d\theta\rangle$ factored over the rational and algebraic factor bases, respectively, and the quadratic characters for $(a+b\theta)\cdot(c+d\theta)$. The goal of §4.4 is to find a non-trivial dependency among the columns of the matrix B, which yields a product of different $a+bm$, $\langle a+b\theta\rangle$, and $a+b\theta$ values that gives a square in $\mathbb{Z}$ and $\mathbb{Z}[\theta]$ by Theorem 3.1.10 and Theorem 3.2.1.

The first bit of $e_{(a,b)}$ is 0 if $a+bm$ is positive and 1 if it is negative, in which case addition modulo 2 of binary vectors $e_{(a,b)}$ and $e_{(c,d)}$ correctly reflects the sign of $(a+bm)\cdot(c+dm)$. The next k bits of $e_{(a,b)}$ are determined by the exponents modulo 2 of every prime in the rational factor base F, when $a+bm$ is factored over F. Similarly, the next l bits of $e_{(a,b)}$ are determined by the exponents modulo 2 of the primes p in first degree prime ideal pairs (r,p) in the algebraic factor base when $\mathrm{N}(a+b\theta)$ is factored over these primes.

Note that if p divides $\mathrm{N}(a+b\theta)$ then there is exactly one (r,p) pair in the algebraic factor base for which $a\equiv -br \pmod{p}$, and that is the (r,p) which is deemed "responsible" for the exponent of the prime p occurring in the factorization of $\mathrm{N}(a+b\theta)$. It's clear that addition modulo 2 of binary vectors $e_{(a,b)}$ and $e_{(c,d)}$ corresponds to the binary vector represented by $(a+bm)\cdot(c+dm)$ and $\langle a+b\theta\rangle\cdot\langle c+d\theta\rangle$ since the latter two multiplications essentially involve addition of exponents.

The final l bits of the vector $e_{(a,b)}$ are determined by each (s,q) pair in the quadratic character base. For a fixed (s,q) pair the corresponding bit in $e_{(a,b)}$ is set to 0 if the Legendre symbol $\left(\frac{a+bs}{q}\right)$ has value 1 and is set to 1 otherwise. This last representation preserves the multiplicative nature of the Legendre symbol in the exact same way the sign of $a+bm$ is preserved during multiplication by the first bit of $e_{(a,b)}$.

A non-trivial dependence among the column vectors of B represents a set U of (a,b) pairs for which the product of the corresponding $a+bm$ values is a square in $\mathbb{Z}$ and for which (3.5) and (3.6) hold. Thus a square has also been produced in $\mathbb{Z}[\theta]$ by §3.2 and Note 2.4.1.

4.4 The Lanczos Algorithm

Given an $n \times n$ matrix A and a column vector y, a standard problem in linear algebra is to find another column vector x such that $Ax = y$. Gaussian elimination is a simple and straightforward method for addressing this problem, but presents difficulties in run-time and storage requirements when n is large. If A possesses the additional properties of being symmetric, positive-definite, and sparse, then the Lanczos method [23] can be employed with dramatic results. The Lanczos algorithm is an iterative method, developed within the context of numerical analysis, for solving the problem of finding the eigensystem of a matrix over $\mathbb{R}$. This method is easily adapted to solving $Ax = y$, and with a little bit of effort can be made to work over the field $\mathbb{F}_2$ of two elements.

To facilitate the exposition, instead of writing the matrix A with coefficients over the field $\mathbb{F}$, the self-adjoint linear operator $T : \mathbb{F}^n \to \mathbb{F}^n$ associated with A will be used. Note that T is self-adjoint since A is assumed to be symmetric. The problem will then be to find a vector $x \in \mathbb{F}^n$ such that $T(x) = y$ for a given vector $y \in \mathbb{F}^n$. In the rest of this exposition the inner product and adjoint operator notation will follow [11, Chapter 6].

In order to motivate the use of a particular subspace in the Lanczos method, recall that if the vector y occurs in the span of an orthogonal set, then a canonical representation for y in terms of the vectors of that set is available:

Theorem 4.4.1. *If $S = \{x_0, x_1, \ldots, x_{n-1}\}$ is an orthogonal set and $y \in \operatorname{span}(S)$, then*

$$y = \frac{\langle y, x_0 \rangle}{\langle x_0, x_0 \rangle} x_o + \frac{\langle y, x_1 \rangle}{\langle x_1, x_1 \rangle} x_1 + \cdots + \frac{\langle y, x_{n-1} \rangle}{\langle x_{n-1}, x_{n-1} \rangle} x_{n-1}$$

Proof. Since $y \in \operatorname{span}(S)$ it follows that $y = a_0 x_0 + a_1 x_1 + \cdots + a_{n-1} x_{n-1}$ with $a_i \in \mathbb{F}$. Then

$$\begin{aligned} \langle y, x_i \rangle &= \langle a_0 x_0 + \cdots + a_i x_i + \cdots + a_{n-1} x_{n-1}, x_i \rangle \\ &= a_0 \langle x_0, x_i \rangle + \cdots + a_i \langle x_i, x_i \rangle + \cdots + a_{n-1} \langle x_{n-1}, x_i \rangle \\ &= a_i \langle x_i, x_i \rangle \end{aligned}$$

by the orthogonality of S. Rewriting this equality as

$$a_i = \frac{\langle y, x_i \rangle}{\langle x_i, x_i \rangle}$$

gives the result. □

At the heart of the Lanczos algorithm is the subspace $W = \operatorname{span}(\{y, T(y), T^2(y), T^3(y), \ldots\})$. In linear algebra circles this is known as the T-cyclic subspace generated by y, while in the realm of numerical analysis it is called the Krylov subspace generated by y. The latter term will be used in this exposition.

Now if a basis $W' = \{w_0, w_1, \ldots, w_{m-1}\}$ for W exists with the property that $\langle w_i, T(w_j)\rangle = 0$ for $i \neq j$, a condition similar to orthogonality, then a canonical representation for a vector corresponding to y can be found, relative to the elements in W', similar to the result given by Theorem 4.4.1. In this case the vector x produced by the representation will not be equal to y but will rather have $T(x) = y$.

Theorem 4.4.2. *If $W' = \{w_0, w_1, \ldots, w_{m-1}\}$ is a basis for the Krylov subspace W generated by y such that $\langle w_i, T(w_j)\rangle = 0$ for $i \neq j$ and $\langle w_i, T(w_i)\rangle \neq 0$ for all $0 \leq i < m$, then the vector*

$$x = \frac{\langle w_0, y\rangle}{\langle w_0, T(w_0)\rangle} w_0 + \frac{\langle w_1, y\rangle}{\langle w_1, T(w_1)\rangle} w_1 + \cdots + \frac{\langle w_{m-1}, y\rangle}{\langle w_{m-1}, T(w_{m-1})\rangle} w_{m-1} \tag{4.2}$$

satisfies $T(x) = y$.

Proof. First we prove that $\langle w, T(x) - y\rangle = 0$ for all $w \in W$. Then

$$\begin{aligned}
\langle w_j, T(x)\rangle &= \left\langle w_j, T\left(\frac{\langle w_0, y\rangle}{\langle w_0, T(w_0)\rangle} w_0 + \cdots + \frac{\langle w_{m-1}, y\rangle}{\langle w_{m-1}, T(w_{m-1})\rangle} w_{m-1}\right)\right\rangle \\
&= \frac{\langle w_0, y\rangle}{\langle w_0, T(w_0)\rangle} \langle w_j, T(w_0)\rangle + \cdots + \frac{\langle w_j, y\rangle}{\langle w_j, T(w_j)\rangle} \langle w_j, T(w_j)\rangle \\
&\quad + \cdots + \frac{\langle w_{m-1}, y\rangle}{\langle w_{m-1}, T(w_{m-1})\rangle} \langle w_j, T(w_{m-1})\rangle \\
&= \langle w_j, y\rangle
\end{aligned}$$

for all $w_j \in W'$. Then $\langle w_j, T(x)\rangle = \langle w_j, y\rangle$ gives $\langle w_j, T(x)\rangle - \langle w_j, y\rangle = 0$ and hence $0 = \langle w_j, T(x) - y\rangle$ for all $w_j \in W'$. But since W' is a basis for W it follows that $\langle w, T(x) - y\rangle = 0$ for all $w \in W$.

Given any $w_i \in W'$ it follows that $\langle w_i, T(T(x) - y)\rangle = \langle T(w_i), T(x) - y\rangle$ since T is self-adjoint. But $T(w_i) \in W$ since W is T-invariant and $0 = \langle T(w_i), T(x) - y\rangle$ since it was just shown that $\langle w, T(x) - y\rangle = 0$ for any $w \in W$. Thus $0 = \langle w_i, T(T(x) - y)\rangle$.

Now since W is T-invariant and $x \in W$ and $y \in W$, it follows that $T(x) - y \in W$ and hence $T(x) - y = c_0 w_0 + \cdots c_{m-1} w_{m-1}$ for some $c_i \in \mathbb{F}$. Then given any $w_i \in W'$

$$\begin{aligned}
0 = \langle w_i, T(T(x) - y)\rangle &= \langle w_i, T(c_0 w_0 + \cdots + c_{m-1} w_{m-1})\rangle \\
&= c_0 \langle w_i, T(w_0)\rangle + \cdots + c_i \langle w_i, T(w_i)\rangle + \cdots + c_{m-1} \langle w_i, T(w_{m-1})\rangle \\
&= c_i \langle w_i, T(w_i)\rangle
\end{aligned}$$

and hence $c_i = 0$ since $\langle w_i, T(w_i)\rangle \neq 0$ by the choice of W'. But i was arbitrary so that $c_i = 0$ for all i with $0 \leq i < m$, and hence $T(x) - y = 0$ and $T(x) = y$. □

As can be seen, the coefficients for the vector x occurring in Theorem 4.4.2 are almost exactly the same as the coefficients occurring in Theorem 4.4.1, except they are adjusted so that the vector x satisfies $T(x) = y$ instead of $x = y$.

This observation leads into the next obvious question, which is exactly how to determine the set W' of basis elements for the Krylov subspace W with the constraint that $\langle w_j, T(w_i)\rangle = 0$ for $i \neq j$. It turns out that this is entirely analogous to the Gram-Schmidt procedure [11, Theorem 6.4] for turning a set of linearly independent vectors into an orthogonal set of (independent) vectors. As will be seen, only slight modification of the coefficients occurring in the Gram-Schmidt process will be needed to produce a basis W' with the aforementioned constraints. But first recall the Gram-Schmidt procedure:

Theorem 4.4.3. *Given a set of linearly independent vectors $S = \{y_0, y_1, \cdots, y_{m-1}\}$, construct the set $S' = \{x_0, x_1, \ldots, x_{m-1}\}$ defined by $x_0 = y_0$ and*

$$x_k = y_k - \frac{\langle y_k, x_0\rangle}{\langle x_0, x_0\rangle} x_0 - \cdots - \frac{\langle y_k, x_{k-1}\rangle}{\langle x_{k-1}, x_{k-1}\rangle} x_{k-1}$$

for $0 < k < m$. Then S' is an orthogonal set and $\operatorname{span}(S) = \operatorname{span}(S')$.

Proof. Let $S_l = \{y_0, y_1, \ldots, y_{l-1}\}$ and $S'_l = \{x_0, x_1, \ldots x_{l-1}\}$ for $0 < l \leq m$. Show S'_l is orthogonal and $\operatorname{span}(S_l) = \operatorname{span}(S'_l)$ using induction on l. The result will then hold for $l = m$.

For $l = 1$ the result is trivially true, so assume $1 \leq l < m$ and the result is true for l. Constructing S'_{l+1} begins with

$$x_l = y_l - \frac{\langle y_l, x_0\rangle}{\langle x_0, x_0\rangle} x_0 - \cdots - \frac{\langle y_l, x_{l-1}\rangle}{\langle x_{l-1}, x_{l-1}\rangle} x_{l-1}$$

First, note that if $x_l = 0$ then $y_l \in \operatorname{span}(S'_l) = \operatorname{span}(S_l)$ which contradicts that S is linearly independent. Hence $x_l \neq 0$. Then for any $x_j \in S'_l$:

$$\begin{aligned}\langle x_l, x_j\rangle &= \left\langle y_l - \frac{\langle y_l, x_0\rangle}{\langle x_0, x_0\rangle} x_0 - \cdots - \frac{\langle y_l, x_j\rangle}{\langle x_j, x_j\rangle} x_j - \cdots - \frac{\langle y_l, x_{l-1}\rangle}{\langle x_{l-1}, x_{l-1}\rangle} x_{l-1}, x_j \right\rangle \\ &= \langle y_l, x_j\rangle - \frac{\langle y_l, x_0\rangle}{\langle x_0, x_0\rangle} \langle x_0, x_j\rangle - \cdots - \frac{\langle y_l, x_j\rangle}{\langle x_j, x_j\rangle} \langle x_j, x_j\rangle - \cdots - \frac{\langle y_l, x_{l-1}\rangle}{\langle x_{l-1}, x_{l-1}\rangle} \langle x_{l-1}, x_j\rangle \\ &= \langle y_l, x_j\rangle - \langle y_l, x_j\rangle = 0\end{aligned}$$

and the orthogonality condition on S'_{l+1} is satisfied. Furthermore $x_l \in \operatorname{span}(S_{l+1})$ and hence $\operatorname{span}(S'_{l+1}) \subseteq \operatorname{span}(S_{l+1})$. But $\dim(\operatorname{span}(S_{l+1})) = \dim(\operatorname{span}(S'_{l+1}))$ and hence $\operatorname{span}(S_{l+1}) = \operatorname{span}(S'_{l+1})$ and the result holds by induction. □

Intuitively, one can think of this process as taking a vector $y_i \in S$ and "orthogonalizing" it against the existing vectors in S' which already orthogonal to one another. This new, orthogonalized vector, is a linear combination of y_i and the existing vectors in S', the latter being linear combinations of previous y_i vectors. With a little bit of adjustment, this procedure can be adapted so as to produce a basis W' satisfying the conditions of Theorem 4.4.2.

At a high level, the process begins with the canonical basis $\{y, T(y), T^2(y), \dots, T^{m-1}(y)\}$ for the Krylov subspace W generated by y. The Gram-Schmidt procedure of Theorem 4.4.3 is then applied to this basis, except with the coefficients adjusted to produce a basis W' which satisfies the hypothesis of Theorem 4.4.2 instead of being orthogonal.

The following result details this sketch and justifies its correctness:

Theorem 4.4.4. *Let* $W = \operatorname{span}(\{y, T(y), T^2(y), \dots, T^{m-1}(y)\})$ *be the Krylov subspace generated by* y *of dimension* m*, and construct the set* $W' = \{w_0, w_1, \dots, w_{m-1}\}$ *with* $w_0 = y$ *and*

$$w_i = T(w_{i-1}) - \frac{\langle T(w_0), T(w_{i-1})\rangle}{\langle T(w_0), w_0\rangle} w_0 - \cdots - \frac{\langle T(w_{i-1}), T(w_{i-1})\rangle}{\langle T(w_{i-1}), w_{i-1}\rangle} w_{i-1} \tag{4.3}$$

Then W' *is a basis for* W *such that* $\langle w_i, T(w_i)\rangle \neq 0$ *and* $\langle w_i, T(w_j)\rangle = 0$ *if* $i \neq j$.

Proof. First, let $W'_l = \{w_0, w_1, \dots, w_{l-1}\}$ be the set with w_i constructed as in (4.3) for $0 \leq i \leq l-1$ and assume the condition that $\langle w_i, T(w_j)\rangle = 0$ for $i \neq j$. It can then be shown that any such set must be linearly independent. To see this, assume $a_0w_0 + \cdots + a_{l-1}w_{l-1} = 0$ for with $a_i \in \mathbb{F}$ for $0 \leq i \leq l-1$. Then $T(a_0w_0 + \cdots + a_{l-1}w_{l-1}) = 0$ and hence for any j with $0 \leq j \leq l-1$ it is seen that

$$\begin{aligned} 0 = \langle w_j, 0\rangle &= \langle w_j, T(a_0w_0 + \cdots + a_{l-1}w_{l-1})\rangle = \langle w_j, a_0T(w_0) + \cdots + a_{l-1}T(w_{l-1})\rangle \\ &= a_0 \langle w_j, T(w_0)\rangle + \cdots + a_j \langle w_j, T(w_j)\rangle + \cdots + a_{l-1} \langle w_j, T(w_{l-1})\rangle = a_j \langle w_j, T(w_j)\rangle \end{aligned}$$

and hence $a_j = 0$ since $\langle w_j, T(w_j)\rangle \neq 0$ and $\langle w_i, T(w_j)\rangle = 0$ for all $i \neq j$ with $0 \leq j \leq l-1$. But since j was arbitrary it follows that the set W'_l must be linearly independent.

As in the proof of Gram-Schmidt, induction will be used on subsets of W and W'. Let $W_l = \{y, T(y), \dots, T^{l-1}(y)\}$ and $W'_l = \{w_0, w_1, \dots, w_{l-1}\}$. The result is trivially true when $l = 1$ so assume the result true for $1 \leq l < m$ and show it holds for $l+1$. First

$$w_l = T(w_{l-1}) - \frac{\langle T(w_0), T(w_{l-1})\rangle}{\langle T(w_0), w_0\rangle} w_0 - \cdots - \frac{\langle T(w_{l-1}), T(w_{l-1})\rangle}{\langle T(w_{l-1}), w_{l-1}\rangle} w_{l-1}.$$

Now for any $w_j \in W'_l$

$$\begin{aligned}
\langle T(w_j), w_l \rangle &= \left\langle T(w_j), T(w_{l-1}) - \frac{\langle T(w_0), T(w_{l-1})\rangle}{\langle T(w_0), w_0\rangle} w_0 - \cdots - \frac{\langle T(w_{l-1}), T(w_{l-1})\rangle}{\langle T(w_{l-1}), w_{l-1}\rangle} w_{l-1} \right\rangle \\
&= \langle T(w_j), T(w_{l-1})\rangle - \frac{\langle T(w_0), T(w_{l-1})\rangle}{\langle T(w_0), w_0\rangle} \langle T(w_j), w_0\rangle - \cdots \\
&\quad - \frac{\langle T(w_j), T(w_{l-1})\rangle}{\langle T(w_j), w_j\rangle} \langle T(w_j), w_j\rangle - \cdots - \frac{\langle T(w_{l-1}), T(w_{l-1})\rangle}{\langle T(w_{l-1}), w_{l-1}\rangle} \langle T(w_j), w_{l-1}\rangle \\
&= \langle T(w_j), T(w_{l-1})\rangle - \langle T(w_j), T(w_{l-1})\rangle \\
&= 0
\end{aligned}$$

and the condition that $\langle w_i, T(w_j)\rangle = 0$ when $i \neq j$ holds for W'_{l+1}.

The next step is to verify that W_{l+1} and W'_{l+1} span the same set. Begin by observing that for any $w_i \in W'$ that $T(w_i) \in \operatorname{span}(\{w_0, w_1, \ldots, w_{i+1}\})$. Now show by induction on k that $w_k = T^k(y) + \sum_{i=0}^{k-1} a_i w_i$ for $a_i \in \mathbb{F}$. For $k = 1$ then

$$w_1 = T(y) - \frac{\langle T(w_0), T(w_0)\rangle}{\langle T(w_0), w_0\rangle} w_0$$

since $w_0 = y$ and this assertion holds. Assuming the result for k, then

$$\begin{aligned}
w_{k+1} &= T(w_k) - \sum_{j=0}^{k} b_j w_j \\
&= T\left(T^k(y) + \sum_{i=0}^{k-1} a_i w_i\right) - \sum_{j=0}^{k} b_j w_j \\
&= T^{k+1}(y) + \sum_{i=0}^{k-1} a_i T(w_i) - \sum_{j=0}^{k} b_j w_j
\end{aligned}$$

with all a_i and b_j in the base field $\mathbb{F}$. The induction follows from the beginning observation about $T(w_i) \in \operatorname{span}(\{w_0, w_1, \ldots, w_{i+1}\})$ since the middle terms in the final summation are all in the span of W'_{k+1}.

All this boils down to showing that $w_l = T^l(y) + \sum_{i=0}^{l-1} a_i w_i$ for some $a_i \in \mathbb{F}$. Now $T^l(y) \in W_{l+1}$ and $w_i \in W_{l+1}$ for $0 \leq i \leq l-1$ since $\operatorname{span}(W_l) = \operatorname{span}(W'_l)$ by the inductive assumption. But then $w_l \in W_{l+1}$ and it follows that $\operatorname{span}(W'_{l+1}) \subseteq \operatorname{span}(W_{l+1})$. But $\dim(\operatorname{span}(W'_{l+1})) = \dim(\operatorname{span}(W_{l+1}))$ so that $\operatorname{span}(W'_{l+1}) = \operatorname{span}(W_{l+1})$ and the result is proved by induction. □

Note that this procedure differs somewhat from the common Gram-Schmidt, in that the vector being orthogonalized is not taken directly from the set S but is rather the vector generated by the previous step of the algorithm. This alteration does not change the validity of the procedure, and in actuality makes the procedure somewhat easier because of the following recurrence relation:

Theorem 4.4.5. *The Lanczos vectors $W' = \{w_0, w_1, \ldots, w_{m-1}\}$ in Theorem 4.4.4 may be computed by the recurrence*

$$w_i = T(w_{i-1}) - \frac{\langle T(w_{i-1}), T(w_{i-1})\rangle}{\langle T(w_{i-1}), w_{i-1}\rangle} w_{i-1} - \frac{\langle T(w_{i-2}), T(w_{i-1})\rangle}{\langle T(w_{i-2}), w_{i-2}\rangle} w_{i-2}$$

for $i \geq 2$.

Proof. Let j be an integer such that $j < i-2$. If it can be shown that $\langle T(w_j), T(w_{i-1})\rangle = 0$ for such j then the result follows. It is immediate that

$$T(w_j) = w_{j+1} + \frac{\langle T(w_0), T(w_j)\rangle}{\langle T(w_0), w_0\rangle} w_0 + \frac{\langle T(w_1), T(w_j)\rangle}{\langle T(w_1), w_1\rangle} w_1 + \cdots + \frac{\langle T(w_j), T(w_j)\rangle}{\langle T(w_j), w_j\rangle} w_j$$

by the form of w_{j+1} in Theorem 4.4.4. It follows that

$$\begin{aligned}\langle T(w_j), T(w_{i-1})\rangle &= \left\langle w_{j+1} + \frac{\langle T(w_0), T(w_j)\rangle}{\langle T(w_0), w_0\rangle} w_0 + \cdots + \frac{\langle T(w_j), T(w_j)\rangle}{\langle T(w_j), w_j\rangle} w_j, T(w_{i-1})\right\rangle \\ &= \langle w_{j+1}, T(w_{i-1})\rangle + \frac{\langle T(w_0), T(w_j)\rangle}{\langle T(w_0), w_0\rangle} \langle w_0, T(w_{i-1})\rangle \\ &\quad + \cdots + \frac{\langle T(w_j), T(w_j)\rangle}{\langle T(w_j), w_j\rangle} \langle w_j, T(w_{i-1})\rangle \\ &= 0\end{aligned}$$

since $j < i-2$ implies that $j+1 < i-1$ and the condition that $\langle w_i, T(w_j)\rangle \neq 0$ in the hypothesis of Theorem 4.4.4. □

In summary, the Lanczos algorithm iteratively constructs a basis W' for the Krylov subspace generated by y that satisfies the hypothesis of Theorem 4.4.2. This allows for the easy computation of a vector x such that $T(x) = y$. In practice, this Lanczos procedure is ideal for large, sparse matrices because the initial matrix is only multiplied by intermediate vectors during each iteration. Among other things, this prevents the performance degradation of methods such as Gaussian elimination which suffer from fill-in of the sparse matrix as the method continues to execute.

4.5 Lanczos in Practice

The ideas described in §4.4 describe how the general Lanczos procedure works, in particular over the field of real numbers $\mathbb{R}$. When this method is adapted to matrices over the field

$\mathbb{Z}/2\mathbb{Z}$, however, other issues must be addressed [23] before a working algorithm can be developed.

The first problem is that the matrix A in §4.4 is assumed to be symmetric, whereas the matrix B of §4.3 is not guaranteed to possess this property. This is remedied by letting $A = B^T B$. Note that if B^T represents an injective linear transformation then a non-trivial vector x which satisfies $A{\cdot}x = 0$ serves as a non-trivial solution to $B{\cdot}x = 0$ since $B^T(B{\cdot}x) = 0$ implies that $B \cdot x = 0$ since B^T is assumed one-to-one. If B^T is not one-to-one, however, the problem still remains as to how to find a non-trivial solution to $B \cdot x = 0$. Fortunately this problem is addressed by a solution to another problem that crops up when adapting the Lanczos method for use in GNFS, outlined below.

As described in §4.3, the goal is to find a dependency among the columns of the matrix B, which amounts to finding a non-trivial vector x such that $B \cdot x = 0$. Unfortunately the method of §4.4 will go absolutely nowhere since the vector y of §4.4 is the zero vector in this case. More specifically, the Krylov subspace generated by the zero vector is the trivial vector space consisting of only the zero vector, and hence the only vector x that will produced by the methods in §4.4 is the trivial vector $x = 0$.

One further problem is that the condition $\langle w_i, T(w_j)\rangle = 0$ for all $i \neq j$ of Theorem 4.4.4 can fail with binary vectors.

To alleviate these difficulties, and to take advantage of the binary nature of the matrix B, in practice the Lanczos method of §4.4 is adapted to a "block" scheme that works with subspaces of vectors instead of individual vectors. First, the matrix A is formed as $A = B^T B$ as alluded to earlier. Next, a random set of vectors represented as columns in a matrix Y is produced and the product AY computed such that AY is not the zero matrix. The set of vectors AY is analogous to y in §4.4. Next, a sequence of subspaces W_i analogous to the w_i vectors in ordinary Lanczos is produced such that there is no vector $w_i \in W_i$ with $\langle w_i, T(w_j)\rangle = 0$ for all $w_j \in W_j$ where $i \neq j$. This latter condition alleviates the difficulty with $\langle w_i, T(w_j)\rangle = 0$ failing. The subspaces W_i are produced with a three-term recurrence similar to the one in Theorem 4.4.5. After enough subspaces have been produced, a set of vectors represented by the matrix X can be found using a formula similar to (4.2) such that $AX = AY$. In this case $A(X - Y) = 0$ and linear combinations of the columns of $X - Y$ may then be computed which produce solutions to $B \cdot x = 0$.

In this block method, all sets and subspaces of vectors are taken to have at most N vectors, where N is the word size of the computer, typically 32 or 64 bits. The native operations of an architecture, such as exclusive or, can in many cases be used to operate on N vectors at a time and hence speed up the algorithm in practice.

4.6 Computing $\phi(\beta)$ When $\beta^2 \in \mathbb{Z}[\theta]$ is Known

Let $\beta \in \mathbb{Z}[\theta]$ be as in Note 2.4.1 and represented as $\beta = a_{d-1}\theta^{d-1} + a_{d-2}\theta^{d-2} + \cdots + a_1\theta + a_0$. Since the natural homomorphism $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/n\mathbb{Z}$ maps $\phi(\theta) = m$ and $\phi(a) \equiv a \pmod{n}$ for $a \in \mathbb{Z}$, let

$$x = a_{d-1}m^{d-1} + a_{d-2}m^{d-2} + \cdots + a_1 m + a_0 \tag{4.4}$$

so that $x \pmod{n}$ produces the difference of squares in Note 2.4.1.

Although it is not practical to compute x directly, x can be computed modulo prime integers p in an easy and efficient manner using finite fields and the techniques of §4.8. The importance of this is justified by the following well-known result [24, Theorem 2.18] of number theory:

Theorem 4.6.1. *Let $p_1, p_2, \ldots, p_k$ be distinct, prime integers and $x_1, x_2, \ldots, x_k$ any sequence of integers. If $P = \prod_{i=1}^{k} p_i$, $P_i = P/p_i$, and $a_i = P_i^{-1} \pmod{p_i}$, then the integer $z = \sum_{i=1}^{k} a_i x_i P_i$ satisfies the congruences*

$$\begin{aligned} z &\equiv x_1 \pmod{p_1} \\ z &\equiv x_2 \pmod{p_2} \\ z &\equiv x_3 \pmod{p_3} \\ &\vdots \\ z &\equiv x_k \pmod{p_k} \end{aligned}$$

simultaneously. Furthermore, z is unique modulo P.

Proof. For each i with $1 \leq i \leq k$, the integer p_i is prime by assumption and hence is relatively prime to P_i since the latter is a product of primes distinct from p_i. This implies that P_i has a multiplicative inverse a_i modulo p_i so that $a_i P_i \equiv 1 \pmod{p}$. But then $a_i x_i P_i \equiv x_i \pmod{p_i}$. Furthermore, if $j \neq i$ it follows that $a_i x_i P_i \equiv 0 \pmod{p_j}$ since p_j divides P_i. It follows immediately that z satisfies the system of congruences as claimed.

To show uniqueness modulo P of z, suppose there is an integer y satisfying this same system of congruences as z. Then $z \equiv y \pmod{p_i}$ for $1 \leq i \leq k$ implies that $z - y$ is divisible by all such p_i. Since each p_i is assumed to be prime it follows that the product P of all these primes must also divide $z - y$ and hence $z \equiv y \pmod{P}$ and uniqueness modulo P follows. □

Applying Theorem 4.6.1 to the situation with x, it is seen that the system of congruences in Theorem 4.6.1 is immediately provided by $x_i = x \pmod{p_i}$ since x can be computed modulo primes p_i easily. If an estimate is available [7, Section 2.2] for the size of x, then enough primes may be chosen such that their product P is larger than x. In that case it follows from the uniqueness of z in Theorem 4.6.1 that $x \equiv z \pmod{P}$ and the value of x becomes evident.

This strategy does avoid the problem of extracting a square root of a polynomials with extremely large coefficients, but it still can encounter insurmountable difficulties. Specifically, each a_i in (4.4) is likely to be enormous by the reasons discussed at the beginning of Chapter 4, so it follows that x itself will be even larger. Even worse is the integer z used in Theorem 4.6.1 to compute x, which will be several orders of magnitude larger than x. Fortunately the problems with the sizes of x and z can be avoided by taking advantage of the fact that $x \equiv z \pmod{P}$ and noting that it is only necessary to compute x modulo n. In fact, these latter two observations allow x to be computed modulo n without ever using any intermediate steps that produce integers of size larger than n.

Begin by noticing that since Theorem 4.6.1 produces an integer z that is much larger than x and such that $x \equiv z \pmod{P}$, it follows that x may be written as $x = z - rP$ where r is the integer

$$r = \left\lfloor \frac{1}{2} + \frac{z}{P} \right\rfloor .$$

The integer r can be though of as the nearest integer to the quotient of z divided by P.

Computing $x \pmod{n}$ is then a matter of computing

$$\begin{aligned} x \pmod{n} &= z \pmod{n} - rP \pmod{n} \\ &= \sum_{i=1}^{k} a_i x_i P_i \pmod{n} - rP \pmod{n}. \end{aligned}$$

Furthermore, r can be found in an efficient manner by noting

$$\frac{z}{P} = \frac{\sum_{i=1}^{k} a_i x_i P_i}{P} = \frac{\sum_{i=1}^{k} a_i x_i \frac{P}{p_i}}{P} = \sum_{i=1}^{k} \frac{a_i x_i}{p_i}$$

and r is computed by rounding the latter expression to the nearest integer.

4.7 Finite Fields and $\mathbb{Q}(\theta)$

Through the use of finite fields, one can easily compute the value $x_i = x \pmod{p_i}$ for prime integers p_i that is necessary for the techniques of §4.6. Begin with the following result which gives a general characterization of all finite fields:

Theorem 4.7.1. *Finite fields $\mathbb{F}_q$ with q elements satisfy the following properties:*

1. *$q = p^d$ for some prime integer p.*

2. *A finite field with $q = p^d$ elements exists for every prime integer p and positive integer d.*

3. *The finite field $\mathbb{F}_q$ is unique.*

4. *The polynomial $x^q - x$ factors as*

$$x^q - x = x \cdot (x - \alpha_1) \cdot (x - \alpha_2) \cdots (x - \alpha_{q-1}) \tag{4.5}$$

in the finite field $\mathbb{F}_q$ for all $\alpha_i \in \mathbb{F}_q^*$.

Proof. Let $\mathbb{F}_q$ be a finite field with q elements. $\mathbb{F}_q$ can not have characteristic 0 since that would imply $\mathbb{F}_q$ is infinite. Hence $\mathbb{F}_q$ must have characteristic p for some prime p and the prime subfield of $\mathbb{F}_q$ must then be $\mathbb{Z}/p\mathbb{Z}$ [10, Chapter 13, Proposition 1]. Since a field forms a vector space in a natural way over any subfield with the subfield taken as the field of scalars [10, page 424], it follows that $\mathbb{F}_q$ is a finite dimensional vector space with field of scalars $\mathbb{Z}/p\mathbb{Z}$, so that

$$\mathbb{F}_q \cong \overbrace{\mathbb{Z}/p\mathbb{Z} \oplus \cdots \oplus \mathbb{Z}/p\mathbb{Z}}^{d}$$

and $\mathbb{F}_q$ has precisely p^d elements.

It's an elementary result from abstract algebra that the non-zero elements of a field form a group under multiplication. In this case, that means the $q-1$ non-zero elements of $\mathbb{F}_q^*$ form a finite group of order $q-1$ under multiplication. Then for any $\alpha \in \mathbb{F}_q^*$ it follows that $\alpha^{q-1} = 1$ and therefore $\alpha^q = \alpha$ for all $\alpha \in \mathbb{F}_q$. Rearranging the last equation yields $\alpha^q - \alpha = 0$ and therefore α is seen to be a root of $x^q - x$ for all $\alpha \in \mathbb{F}_q$. This determines q roots of $x^q - x$ and since $x^q - x$ has at most q roots it follows that $\mathbb{F}_q$ consists of exactly the roots of $x^q - x$. It follows that $\mathbb{F}_q$ is the splitting field for $x^q - x$. From the existence and uniqueness of splitting fields [10, Chapter 13, Theorem 25 and Corollary 28] it follows that finite fields exist and are unique.

Now since the roots of $x^q - x$ are determined precisely by the elements $\alpha \in \mathbb{F}_q$, it follows that the linear factors of $x^q - x$ are determined precisely by linear polynomials $(x - \alpha)$ associated with each $\alpha \in \mathbb{F}_q$. Then (4.5) follows immediately. □

The next result serves as an analog of Theorem 2.2.2 when a polynomial is available that is irreducible over $\mathbb{Z}/p\mathbb{Z}$ instead of $\mathbb{Q}$, and also provides for a convenient representation for a finite field:

Theorem 4.7.2. *Let $f(x)$ be monic polynomial of degree d with integer coefficients which is irreducible over $\mathbb{Z}/p\mathbb{Z}$ for some prime integer p. If θ_p denotes a root of $f(x)$ in the splitting field for $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$ then*

1. *$\mathbb{F}_p[x]/(f(x))$ is the finite field $\mathbb{F}_q$ with $q = p^d$ elements*
2. *$f(x)$ divides any polynomial $g(x)$ over $\mathbb{Z}/p\mathbb{Z}$ for which $g(\theta_p) \equiv 0 \pmod{p}$.*
3. *Every element of $\mathbb{F}_q$ may be expressed as $\mathbb{Z}/p\mathbb{Z}$-linear combinations of the elements*

$$\{1, \theta_p, \theta_p^2, \ldots, \theta_p^{d-1}\}$$

and hence $\mathbb{F}_q \cong \mathbb{F}_p(\theta_p)$ where $\mathbb{F}_p(\theta_p)$ denotes the ring of all polynomials in θ_p with coefficients modulo p.

Proof. The proof of Theorem 2.2.2 applies here with the field $\mathbb{Z}/p\mathbb{Z}$ substituted for the field $\mathbb{Q}$. Note that no special properties of $\mathbb{Q}$ or any field of characteristic 0 is used in the proof of Theorem 2.2.2, only the assumption that $\mathbb{Q}$ is a field is essential. □

For the purposes of the GNFS, the importance of Theorem 4.7.2 is that it provides a natural correspondence between the finite field with p^d elements (for some prime p) and the field $\mathbb{Q}(\theta)$ when the polynomial $f(x)$ is irreducible over both $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Q}$. More specifically there is a natural ring epimorphism $\tau_p : \mathbb{Z}[\theta] \to \mathbb{F}_p(\theta_p)$ with $\tau_p(\theta) = \theta_p \pmod{p}$ and $\tau_p(a) \equiv a \pmod{p}$ for all $a \in \mathbb{Z}$, which simply reduces the coefficients of polynomials in $\mathbb{Z}[\theta]$ modulo p.

Now if $\beta^2 = \delta \in \mathbb{Z}[\theta]$ as in §4.6 then

$$\delta_p = \tau_p(\delta) = \tau_p(\beta^2) = (\tau_p(\beta))^2 = \beta_p^2$$

in $\mathbb{F}_p(\theta_p)$. Thus, a square root in $\mathbb{F}_p(\theta_p)$ of δ is equivalent to $\pm\beta$ in $\mathbb{F}_p(\theta_p)$, so that in particular β_p can be thought of as β with its coefficients reduced modulo p. But since reducing the coefficients of β modulo p does not effect the computation of x in (4.4) modulo p, it follows that x can be computed modulo p by substituting m in for θ_p in the representation of β_p as a polynomial in θ_p and reducing modulo p. Thus, values $x_i = x \pmod{p_i}$ may be computed easily assuming that β_p may be calculated from δ_p in $\mathbb{F}_p(\theta_p)$, which turns the problem into one of extracting square roots in a finite field efficiently. The latter issue is addressed in §4.8.

It could so happen that for primes p_i and p_j for which $f(x)$ is irreducible that different square roots of δ are computed in the fields $\mathbb{F}_{p_i}(\theta_{p_i})$ and $\mathbb{F}_{p_i}(\theta_{p_j})$. More explicitly, it could be that β_{p_i} is computed in one field and $-\beta_{p_j}$ and the other, both of which are valid square roots of δ in the respective fields. The consequence of this happening is that $x_i = x \pmod{p_i}$ and $x_j = -x \pmod{p_j}$, but the latter is treated as $x_j = x \pmod{p_j}$ by the methods of §4.6. Then z will be computed with $z \equiv x_j \pmod{p_j}$ when it should be that $z \equiv -x_j \pmod{p_j}$ and hence $z \not\equiv x \pmod{P}$ and the method breaks down. To avoid this scenario when computing square roots of δ in different finite fields, care should be taken that the square roots are all simultaneously equivalent to either β or $-\beta$.

At this point, the assumption that the degree of the defining polynomial $f(x)$ be odd comes into play:

Theorem 4.7.3. *Let $f(x)$ be a monic, irreducible polynomial of odd degree d with integer coefficients. Then for any $\alpha \in \mathbb{Q}(\theta)$ it follows that* $\mathrm{N}(-\alpha) = -\mathrm{N}(\alpha)$.

Proof. Using the embeddings $\sigma_i : \mathbb{Q}(\theta) \to \mathbb{C}$ from Theorem 3.1.2 it follows that

$$\mathrm{N}(-\alpha) = \sigma_1(-\alpha) \cdot \sigma_2(-\alpha) \cdots \sigma_d(-\alpha) = (-1)^d \sigma_1(\alpha) \cdot \sigma_2(\alpha) \cdots \sigma_d(\alpha) = -\mathrm{N}(\alpha)$$

since d is odd. □

From Theorem 4.7.3 it follows that either β or $-\beta$ has positive norm, so without loss of generality assume β does.

One can ensure then that for every finite field $\mathbb{F}_p(\theta_p)$ in which a square root β_p is computed that β_p is equivalent to β and not $-\beta$ by determining the norm of the element that β_p is equivalent to. If $-\beta_p$ happened to be computed then negating the coefficients in its representation as a polynomial in $\mathbb{F}_p(\theta_p)$ will yield the desired β_p.

The remaining question then is how to determine the norm of the element that a particular square root in $\mathbb{F}_p(\theta_p)$ represents. The answer is that the norm function of N of Definition 3.1.1 has a counterpart N_p for the finite field $\mathbb{F}_p(\theta_p)$ such that $\mathrm{N}(\alpha) \equiv \mathrm{N}_p(\alpha) \pmod{p}$ for all $\alpha \in \mathbb{Z}[\theta]$. This is adequate for the GNFS, because the actual value of the norm is not important, only its sign.

As in the proof of Theorem 4.7.2, if the polynomial $f(x)$ is irreducible over $\mathbb{Z}/p\mathbb{Z}$ then $f(x)$ divides any polynomial $g(x)$, modulo p, which shares a root with $f(x)$. From Theorem 4.7.2 the finite field $\mathbb{F}_q$ with $q = p^d$ elements can be represented as elements of $\mathbb{F}_p(\theta_p)$ where θ_p is a root of $f(x)$ in the splitting field of $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$. Then $\theta_p \in \mathbb{F}_q$ implies that θ_p is a root of $x^q - x$ by Theorem 4.7.1. Hence $f(x)$ divides $x^q - x$ and since $x^q - x$ factors into distinct, linear factors by Theorem 4.7.1 it follows that $f(x)$ does as well. Hence $f(x)$ has d distinct roots in the splitting field of $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$. Furthermore, the splitting field $\mathbb{F}_q$ of $x^q - x$ contains all these roots of $f(x)$. Thus, the embeddings of Theorem 3.1.2 carry over to the finite $\mathbb{F}_q$, in that there are exactly d automorphisms defined on $\mathbb{F}_q$, each of which sends θ_p to a distinct root of $f(x)$ in $\mathbb{F}_q$. This allows for the concept of a norm function in $\mathbb{F}_q$ to be defined:

Definition 4.7.1. Let $f(x)$ be a monic polynomial of degree d with integer coefficients that is irreducible modulo p for some prime integer p. If θ_p is a root of $f(x)$ in the splitting field of $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$, then the *norm* of an element α in the finite field $\mathbb{F}_q \cong \mathbb{F}_p(\theta_p)$ for $q = p^d$ is defined to be

$$\mathrm{N}_p(\alpha) = \sigma_1(\alpha)\sigma_2(\alpha) \cdots \sigma_d(\alpha) \tag{4.6}$$

where the σ_i are the distinct automorphisms of $\mathbb{F}_q$ which map θ_p to the d distinct roots of $f(x)$ in $\mathbb{F}_q$.

Because of the special structure of the finite field $\mathbb{F}_q$, the embeddings σ_i of Definition 4.7.1 have a natural representation that allows for easy calculation of (4.6). Specifically:

Theorem 4.7.4. *The group of automorphisms of the finite field $\mathbb{F}_q$ where $q = p^d$ forms a cyclic group, generated by the Frobenius automorphism σ_p, where $\sigma_p(\alpha) = \alpha^p$ for $\alpha \in \mathbb{F}_q$.*

Proof. First, show that σ_p is an automorphism of $\mathbb{F}_q$. Let $\alpha \in \mathbb{F}_q$ and $\beta \in \mathbb{F}_q$. Then

$$\sigma_p(\alpha \cdot \beta) = (\alpha \cdot \beta)^p = \alpha^p \cdot \beta^p = \sigma_p(\alpha) \cdot \sigma_p(\beta)$$

so the multiplicative structure of $\mathbb{F}_q$ is preserved by σ_p. Also,

$$\begin{aligned}\sigma_p(\alpha + \beta) = (\alpha + \beta)^p = \alpha^p &+ \binom{p}{1}\alpha^{p-1}\beta + \binom{p}{2}\alpha^{p-2}\beta^2 \\ &+ \cdots + \binom{p}{i}\alpha^{p-i}\beta^i + \cdots + \binom{p}{p-1}\alpha\beta^{p-1} + \beta^p \\ = \alpha^p + \beta^p = \sigma_p(\alpha) + \sigma_p(\beta)\end{aligned}$$

since the binomial coefficients occurring in the middle terms of the expansion of $(\alpha + \beta)^p$

$$\binom{p}{i} = \frac{p!}{(p-i)!\,i!}$$

are a multiples of p and hence equivalent to 0 in $\mathbb{F}_q$, since $i < p$ and $p-i < p$ for $1 \le i \le p-1$.

To show that σ_p is one-to-one, suppose $\sigma_p(\alpha) = \alpha^p = 0$. Then $\alpha^{p^d} = 0$. But $\alpha^{p^d} = \alpha$ from Theorem 4.7.1 and hence $\alpha^{p^d} = 0$ implies that $\alpha = 0$ and σ_p must be injective. Since σ_p is injective and maps a finite set to itself it follows by the pigeon hole principal that σ_p is onto and hence an isomorphism.

Since the automorphisms of $\mathbb{F}_q$ form a group under composition, each σ_p^k is also an automorphism for any integer k. Now $\sigma_p^2(\alpha) = \sigma_p(\sigma_p(\alpha)) = \sigma_p(\alpha^p) = (\alpha^p)^p = \alpha^{p^2}$, and similarly $\sigma_p^k(\alpha) = \alpha^{p^k}$ for any integer k. From Theorem 4.7.1 it is known that $\alpha^{p^d} = \alpha$ for all $\alpha \in \mathbb{F}_q$, hence σ_p^d is the identity automorphism of $\mathbb{F}_q$. Suppose there is an integer $k < d$ for which σ_p^k is the identity automorphism. Then $\alpha^{p^k} = \alpha$ for all $\alpha \in \mathbb{F}_q$. But this implies that the polynomial $x^{p^k} - x$ has p^d roots in $\mathbb{F}_q$, which can't happen if $k < d$. Hence σ_p generates a group of automorphisms of $\mathbb{F}_q$ of order d, and since there are only d automorphism of $\mathbb{F}_q$, it follows that the automorphism group of $\mathbb{F}_q$ is cyclic and generated by σ_p. □

Corollary. *The norm of an element α in the finite field $\mathbb{F}_q$ with $q = p^d$ may be computed as*

$$\mathrm{N}_p(\alpha) = \alpha^{\frac{p^d-1}{p-1}}.$$

Proof. From the main theorem and Definition 4.7.1 it follows that

$$\begin{aligned}\mathrm{N}_p(\alpha) &= \sigma_1(\alpha)\sigma_2(\alpha)\cdots\sigma_d(\alpha) = \sigma_p(\alpha)\sigma_p^2(\alpha)\cdots\sigma_p^d(\alpha) = \alpha^p\cdot\alpha^{p^2}\cdots\alpha^{p^d}\\ &= \alpha^{1+p+p^2+\cdots+p^{d-1}} = \alpha^{\frac{p^d-1}{p-1}}\end{aligned}$$

since every automorphism σ_i of $\mathbb{F}_q$ is a power of the Frobenius automorphism $\sigma_p(\alpha)=\alpha^p$. □

4.8 Computing Square Roots in $\mathbb{F}_{p^d}$

The method of Shanks and Tonelli [16, Section 9.2] for finding square roots of integers modulo primes is immediately applicable to finite fields $\mathbb{F}_q$ with $q=p^d$ elements, for p prime and d positive. This follows because the basic assumptions and operations of that algorithm apply to any cyclic group of even order, and the multiplicative group $\mathbb{F}_q^*$ of the $q-1$ nonzero elements of any finite field with q odd satisfies those requirements.

Given a perfect square $\delta\in\mathbb{F}_q^*$ and a generator γ for $\mathbb{F}_q^*$, there are three basic methods for finding $\nu\in\mathbb{F}_q^*$ with $\nu^2=\delta$. Since $\mathbb{F}_q^*$ has a finite number of elements, the brute-force method proceeds by examining every power of γ in the group $\mathbb{F}_q^*$ until the square is found. This could take as many as $(q-1)/2$ steps, which quickly grows unwieldy as the size of q increases.

A second method increases efficiency by narrowing down considerably the possible elements of $\mathbb{F}_q^*$ whose square could be δ. First, the notions of quadratic residues and non-residues are generalized from the integers to $\mathbb{F}_q^*$:

Definition 4.8.1. Given a finite field $\mathbb{F}_q$ with $q=p^d$ elements and p an odd, prime integer, an element $\delta\in\mathbb{F}_q^*$ is called a *quadratic residue in* $\mathbb{F}_q^*$ if there is an element $\nu\in\mathbb{F}_q^*$ such that $\nu^2=\delta$ and is called a *quadratic non-residue in* $\mathbb{F}_q^*$ otherwise.

A version of Euler's Criterion then follows cleanly:

Theorem 4.8.1. *Let $\mathbb{F}_q$ be a finite field with $q=p^d$ elements where p is an odd, prime integer. An element $\delta\in\mathbb{F}_q^*$ is a quadratic residue in $\mathbb{F}_q^*$ if and only if $\delta^{(q-1)/2}=1$ and is a quadratic non-residue in $\mathbb{F}_q^*$ if and only if $\delta^{(q-1)/2}=-1$*

Proof. Let $\gamma\in\mathbb{F}_q^*$ be a generator for $\mathbb{F}_q^*$, and suppose δ is a quadratic residue in $\mathbb{F}_q^*$ so that $\delta=(\gamma^k)^2=\gamma^{2k}$ for some integer k. Then

$$\delta^{\frac{(q-1)}{2}} = \gamma^{\frac{2k(q-1)}{2}} = (\gamma^{(q-1)})^k = 1^k = 1$$

since the order of $\mathbb{F}_q^*$ is $q-1$.

If instead δ is a quadratic non-residue in $\mathbb{F}_q^*$ then $\delta = \gamma^{2k+1}$ for some k (δ can't be an even power of γ for if it were it would be a quadratic residue in $\mathbb{F}_q^*$). Note that $\gamma^{(q-1)/2} = -1$ since $\gamma^{(q-1)} = 1$ and γ has order $q-1$. Then

$$\delta^{\frac{(q-1)}{2}} = \gamma^{\frac{(2k+1)(q-1)}{2}} = \gamma^{\frac{2k(q-1)}{2}} \cdot \gamma^{\frac{(q-1)}{2}} = (\gamma^{(q-1)})^k \cdot \gamma^{\frac{(q-1)}{2}} = 1 \cdot (-1) = -1$$

Note that $\delta^{(q-1)/2} = \pm 1$ since the order of $\mathbb{F}_q^*$ is $q-1$ and hence $\delta^{(q-1)} = 1$. This observation forces the converses of the above two statements. □

To begin to improve upon the brute-force method, let $\delta \in \mathbb{F}_q^*$ be a quadratic residue in $\mathbb{F}_q^*$ and factor $q-1 = 2^r \cdot s$ where s is odd (possibly 1) and $r > 0$ since q is assumed odd. Note that if $\omega = \delta^{(s+1)/2}$ then $\omega^2 = \delta^s \cdot \delta$ and hence δ^s is a quadratic residue in $\mathbb{F}_q^*$. Thus there exists $\zeta \in \mathbb{F}_q^*$ such that $\zeta^2 = \delta^s$. But then letting $\nu = \omega \cdot \zeta^{-1}$ it follows that

$$\nu^2 = \omega^2 \cdot \zeta^{-2} = \delta^s \cdot \delta \cdot \delta^{-s} = \delta$$

and a square root of δ has been produced.

The question then becomes one of finding the element ζ, which turns out to be much easier than searching all of $\mathbb{F}_q^*$ by the following result:

Theorem 4.8.2. *Let $\mathbb{F}_q$ be a finite field with $q = p^d$ elements where p is an odd, prime integer. If $q-1 = 2^r \cdot s$ and $\delta \in \mathbb{F}_q^*$ is a quadratic residue in $\mathbb{F}_q^*$, then the element $\delta^s \in \mathbb{F}_q^*$ has order dividing 2^{r-1} in $\mathbb{F}_q^*$.*

Proof. From basic group theory [14, Chapter 1, Theorem 3.4], the order of the element $\delta \in \mathbb{F}_q^*$ divides any power k of δ for which $\delta^k = 1$. From the assumption that δ is a quadratic residue in $\mathbb{F}_q^*$ and Theorem 4.8.1 it follows that $\delta^{(q-1)/2} = 1$. Then

$$(\delta^s)^{2^{r-1}} = \delta^{2^{r-1}s} = \delta^{\frac{2^r s}{2}} = \delta^{\frac{q-1}{2}} = 1$$

and hence the order of δ^s must divide 2^{r-1} by the initial comment. □

Corollary. *If $\zeta \in \mathbb{F}_q^*$ with $z^2 = \delta^s$, then ζ has order dividing 2^r.*

Proof. Let k denote the order of δ^s in $\mathbb{F}_q^*$, so that k must divide 2^{r-1}. Then k also denotes the order of ζ^2 so that $1 = (\zeta^2)^k = \zeta^{2k}$ and hence the order of ζ must divide $2k$, which in turn must divide 2^r since k divides 2^{r-1}. Thus the order of z divides 2^r by transitivity. □

Given that every element of the Sylow 2-subgroup S_{2^r} of $\mathbb{F}_q^*$ has order dividing 2^r, and furthermore that any element of $\mathbb{F}_q^*$ having order dividing 2^r is also an element of S_{2^r}, it follows that $\zeta \in S_{2^r}$. The problem of finding a square root of δ then has been reduced from searching the $q-1$ elements of $\mathbb{F}_q^*$ to only searching the 2^r elements in S_{2^r}. In addition, a very convenient representation for S_{2^r} is derived from the following result:

Theorem 4.8.3. *Let $\mathbb{F}_q$ be a finite field with $q = p^d$ elements where p is an odd, prime integer. If $\eta \in \mathbb{F}_q^*$ is a quadratic non-residue in $\mathbb{F}_q^*$ then η^s has order 2^r. In particular, the Sylow 2-subgroup S_{2^r} of $\mathbb{F}_q^*$ is given by*

$$S_{2^r} = \{1, n^s, n^{2s}, n^{3s}, \ldots, n^{(2^r-1)s}\}.$$

Proof. Let k denote the order of η^s and note $\eta^{(q-1)/2} = -1$ by Theorem 4.8.1 since η is assumed to be a quadratic non-residue in $\mathbb{F}_q^*$. Then

$$-1 = \eta^{\frac{(q-1)}{2}} = \eta^{\frac{2^r s}{2}} = \eta^{2^{r-1}s} = (\eta^s)^{2^{r-1}}$$

so that $(\eta^s)^{2^r} = 1$ and therefore k must divide 2^r. Now $(\eta^s)^{2^m} \neq 1$ for $0 \leq m < r-1$ since otherwise

$$(\eta^s)^{2^{m+1}} = (\eta^s)^{2^{m+2}} = \cdots (\eta^s)^{2^{r-1}} = 1$$

and it is known that $(\eta^s)^{2^{r-1}} = -1$. Since k must divide 2^r and $k \neq 2^m$ for $0 \leq m < r$ it follows that $k = 2^r$ as desired.

Since η^s has order 2^r it generates a subgroup of $\mathbb{F}_q^*$ of order 2^r. But there is only one Sylow 2-subgroup S_{2^r} of $\mathbb{F}_q^*$ so η^s must generate S_{2^r} and the result follows. □

The method of Shanks and Tonelli improves even further upon this, requiring at most r steps instead of 2^r. The idea is to produce a sequence of elements ω_i and λ_i in $\mathbb{F}_q^*$ such that $\omega_i^2 = \lambda_i \delta$, with the order o_{i+1} of λ_{i+1} strictly less than the order o_i of λ_i and o_i dividing 2^{r-1} for all i in the sequence. If such a sequence could be found then eventually $\lambda_j = 1$ for some j so $\omega_j^2 = a$ and a square root of δ has been found. Note in the worst case $o_0 = 2^{r-1}$, $o_1 = 2^{r-2}$, ... , $o_{r-1} = 1$ and a total of r steps is required to find the square root, which is significantly better than the 2^r potential steps required when examining the elements of the Sylow 2-subgroup S_{2^r}.

The overall technique then is to produce the sequence of λ_i's whose order satisfies the mentioned conditions, then to derive the ω_i's from the equation $\omega_i^2 = \lambda_i \delta$. Begin by letting $\lambda_0 = \delta^s$ and $\omega_0 = \delta^{(s+1)/2}$. The following result then details how to choose the remaining λ_i:

Theorem 4.8.4. *Let ζ be a generator for the Sylow-2 subgroup S_{2^r}. If λ_i has order $o_i = 2^m$ then $\lambda_{i+1} = \lambda_i \zeta^{2^{r-m}}$ has order o_{i+1} dividing 2^{m-1} and hence $o_{i+1} < o_i$.*

Proof. Since the order o_i of λ_i is 2^m then $\lambda_i^{2^m} = 1$ and using similar logic from the proof of Theorem 4.8.3 it follows that $\lambda_i^{2^{m-1}} = -1$. Similarly, since ζ generates S_{2^r} and hence has order 2^r, it follows that $\zeta^{2^{r-1}} = -1$. Then

$$\lambda_{i+1}^{2^{m-1}} = \lambda_i^{2^{m-1}} z^{2^{(r-m)+(m-1)}} = \lambda_i^{2^{m-1}} \zeta^{2^{r-1}} = (-1)(-1) = 1$$

and hence o_{i+1} divides 2^{m-1}. □

Note from Theorem 4.8.3 that $\zeta = \eta^s$ may be taken as a generator for S_{2^r}, where η is a quadratic non-residue in $\mathbb{F}_q^*$.

The following result completes the algorithm by describing how to compute the ω_i corresponding to λ_i:

Theorem 4.8.5. *If ζ is a generator for the Sylow-2 subgroup S_{2^r}, λ_i has order 2^m, and ω_i satisfies $\omega_i^2 = \lambda_i\delta$ then $\omega_{i+1} = \omega_i\zeta^{2^{r-m-1}}$ satisfies $\omega_{i+1}^2 = \lambda_{i+1}\delta$ where λ_{i+1} is chosen as in Theorem 4.8.4.*

Proof. From Theorem 4.8.4 it follows that λ_{i+1} is of the form $\lambda_{i+1} = \lambda_i\zeta^{2^{r-m}}$. Then

$$\omega_{i+1}^2 = \omega_i^2 \cdot (\zeta^{2^{r-m-1}})^2 = \omega_i^2 \cdot \zeta^{2^{r-m}} = \lambda_i \cdot \delta \cdot \zeta^{2^{r-m}} = \lambda_{i+1}^2\delta$$

and the result holds. □

4.9 Irreducibility Testing of Polynomials Modulo p

In order to develop an efficient method for determining if a monic polynomial of degree d is irreducible over $\mathbb{Z}/p\mathbb{Z}$ for a prime integer p, we consider some theory about the subfield structure of the finite field $\mathbb{F}_{p^d}$. Understanding this structure will lead immediately to an understanding of the monic irreducible polynomials over $\mathbb{Z}/p\mathbb{Z}$ and hence to a method for testing irreducibility.

Lemma 4.9.1. *Let a and b be positive integers such that a divides b. Then the polynomial $x^a - 1$ divides the polynomial $x^b - 1$.*

Proof. Since a and b are positive integers with a dividing b, it follows that b may be written as $b = ak$ for some integer $k \geq 1$. Then

$$\begin{aligned}(x^a - 1)\cdot(x^{a(k-1)} + x^{a(k-2)} + x^{a(k-3)} + \cdots + x^a + 1) &= x^{ak} + x^{a(k-1)} + x^{a(k-2)} \\ + \cdots + x^{2a} + x^a - x^{a(k-1)} - x^{a(k-2)} - \cdots - x^{2a} - x^a - 1 &= x^{ak} - 1 = x^b - 1\end{aligned}$$

so indeed $x^a - 1$ divides $x^b - 1$. □

Theorem 4.9.2. *Given a prime integer p and a positive integer d, let $\mathbb{F}_{p^d}$ denote the finite field containing p^d elements. Any subfield of $\mathbb{F}_{p^d}$ contains p^e elements where e is a divisor of d. Conversely, if e is a divisor of d then there is exactly one subfield of $\mathbb{F}_{p^d}$ containing p^e elements.*

Proof. Suppose E is a subfield of $\mathbb{F}_{p^d}$. Then $d = [\mathbb{F}_{p^d} : \mathbb{F}_p] = [\mathbb{F}_{p^d} : E][E : \mathbb{F}_p]$ and hence E is a finite dimensional extension of $\mathbb{F}_p$ and must therefore have p^e elements for some positive integer e. But this equation also implies that $\mathbb{F}_{p^d}$ is a finite dimensional extension of E and hence $\mathbb{F}_{p^d}$ must contain $(p^e)^k$ elements for some positive integer k. But $\mathbb{F}_{p^d}$ contains p^d elements so $p^d = p^{ek}$ and therefore d is a multiple of e.

Conversely, suppose e divides d. Then applying Lemma 4.9.1 with $a = e$, $b = d$, and $x = p$ it follows that $p^e - 1$ divides $p^d - 1$. Since $p^e - 1$ divides $p^d - 1$ Lemma 4.9.1 can be applied again with $a = p^e - 1$ and $b = p^d - 1$ to see that $x^{p^e-1} - 1$ divides $x^{p^d-1} - 1$ and hence $x^{p^e} - x$ divides $x^{p^d} - x$. The latter result implies that every root of $x^{p^e} - x$ is also a root of $x^{p^d} - x$ and hence the splitting field for the latter polynomial contains as a subfield the splitting field for the former. By Theorem 4.7.1 the splitting field for $x^{p^d} - x$ is $\mathbb{F}_{p^d}$ and the splitting field for $x^{p^e} - x$ is $\mathbb{F}_{p^e}$ and hence $\mathbb{F}_{p^d}$ has $\mathbb{F}_{p^e}$ as a subfield, the latter containing p^e elements. Now if $\mathbb{F}_{p^d}$ contained more than one subfield with p^e elements then there would be more than p^e elements in $\mathbb{F}_{p^d}$ that satisfy the polynomial $x^{p^e} - x$. The latter is impossible so that $\mathbb{F}_{p^d}$ is seen to have exactly one subfield with p^e elements. □

The following result about irreducibles justifies the concern about the subfield structure of $\mathbb{F}_{p^d}$:

Theorem 4.9.3. *Let $f(x)$ be a monic, irreducible polynomial of degree e over $\mathbb{Z}/p\mathbb{Z}$. Then $f(x)$ divides $x^{p^d} - x$ if and only if e divides d.*

Proof. Suppose $f(x)$ divides $x^{p^d} - x$. If α is a root of $f(x)$ in the splitting field of $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$ then $f(\alpha) = 0$ implies that $\alpha^{p^d} - \alpha = 0$ and hence $\alpha \in \mathbb{F}_{p^d}$. Thus $\mathbb{F}_p(\alpha)$ is a subfield of $\mathbb{F}_{p^d}$ which contains p^e elements by Theorem 4.7.2 since $f(x)$ is irreducible over $\mathbb{Z}/p\mathbb{Z}$. From Theorem 4.9.2 it follows that e must divide d.

Conversely, suppose e divides d. Then by Theorem 4.9.2 $\mathbb{F}_{p^d}$ contains a subfield with p^e elements. But $\mathbb{F}_p(\alpha)$ is just such a field by Theorem 4.7.2. Then $\alpha \in \mathbb{F}_{p^d}$ so that $\alpha^{p^d} - \alpha = 0$ and hence α is a root of $x^{p^d} - x$. But $f(x)$ divides any polynomial that has α as a root by Theorem 4.7.2, so in particular $f(x)$ must divide $x^{p^d} - x$. □

One final result encapsulates all the information needed about irreducibles to develop a good test for irreducibility:

Theorem 4.9.4. *The polynomial $x^{p^d} - x$ is the product of all distinct, monic, irreducible polynomials over $\mathbb{Z}/p\mathbb{Z}$ whose degree divides d.*

Proof. From Theorem 4.9.3 it follows that when $x^{p^d} - x$ is factored into irreducibles over $\mathbb{Z}/p\mathbb{Z}$ that each of these irreducibles has degree dividing d, and every irreducible of such degree occurs as a factor of $x^{p^d} - x$. Now the derivative of $x^{p^d} - x$ is -1 modulo p, so that $x^{p^d} - x$ and its derivative share no roots. Hence $x^{p^d} - x$ is separable [14, page 261] and therefore none of the irreducibles in its unique factorization into irreducibles are repeated. □

A procedure for determining irreducibility [6, Algorithm 1.3.14] is summarized in the following result:

Theorem 4.9.5. *A monic polynomial $f(x)$ over $\mathbb{Z}/p\mathbb{Z}$ of degree d is irreducible if and only if $f(x)$ divides $x^{p^d} - x$ and $\gcd(x^{d/p_i} - x, f(x)) = 1$ for all primes p_i dividing d.*

Proof. Suppose $f(x)$ is irreducible. Then $f(x)$ must divide $x^{p^d} - x$ by Theorem 4.9.3. Since $f(x)$ is irreducible it has no non-trivial factors and the gcd condition follows.

Conversely, assume $\gcd(x^{d/p_i} - x, f(x)) = 1$ for all primes p_i dividing d. Suppose that $f(x)$ is reducible. Then $f(x)$ has some non-trivial, irreducible factor $g(x)$ of degree e such that $0 < e < d$. Since $g(x)$ divides $f(x)$ and $f(x)$ divides $x^{p^d} - x$ it follows that $g(x)$ must divide $x^{p^d} - x$ and hence e divides d by Theorem 4.9.3. If $d = p_1^{a_1} \cdots p_k^{a_k}$ denotes the unique factorization of d into distinct primes p_i, then $e = p_1^{b_1} \cdots p_k^{b_k}$ and $0 \leq b_i \leq a_i$ for $1 \leq i \leq k$. But there exists at least one $b_j < a_j$ since $e < d$ and hence e divides $p_1^{a_1} \cdots p^{a_j - 1} \cdots p_k^{a_k} = d/p_i$. But then $g(x)$ must divide $x^{p^{d/p_i}} - x$ by Theorem 4.9.3 and since $g(x)$ divides $f(x)$ it follows that $\gcd(x^{d/p_i} - x, f(x)) \neq 1$, a contradiction. Thus, $f(x)$ has no non-trivial factors and is hence irreducible. □

Chapter 5

An Extended Example

In order to concretize the concepts of the GNFS outlined in the earlier sections, it is helpful to see an example that illustrates the different stages of the algorithm. This goal will be achieved by working through the factorization of the integer $45,113$ using the GNFS. Although numbers of such magnitude would never be factored with GNFS in practice, this particular integer serves as a useful example since the sizes of the factor bases can be kept small and intermediate computations can be detailed without becoming unwieldy.

The choice of d, m, and the polynomial $f(x)$ for $45,113$ are detailed in §5.1. A word about a representation for the rational factor base is given in §5.2, while the algebraic factor base, quadratic character base and finding roots of $f(x)$ modulo p for various primes p is detailed in §5.3 and §5.4. Some examples of the sieving process are given in §5.5, as well as the (a, b) pairs that produce smooth values for $a + bm$ and $a + b\theta$. Having enough of the latter (a, b) pairs leads to the linear algebra step discussed in §5.6 and §5.7.

Since the integer $45,113$ is relatively small, an explicit square root in $\mathbb{Z}[\theta]$ is computed in §5.8, although this is never done in practice. It is useful in this example, though, to check the validity of the techniques in §5.9, §5.10, and §5.11.

The final difference of squares produced for this example is detailed in §5.12 and a non-trivial factorization of $45,113$ is revealed.

5.1 Selecting the Polynomial

The first parameter to decide upon for factoring $n = 45,113$ is the degree d of the polynomial $f(x)$ that will drive the rest of the algorithm. From the remarks made in §4.1 and the requirement that d be odd it is decided that $d = 3$ will be used for this particular n. Next, m should be chosen with $m \approx n^{1/d}$, which in this case indicates m should be around $45,113^{1/3} \approx 35$. Although $m = 35$ would yield a monic polynomial following the base-m

method of §4.1, the value $m = 31$ serves equally well and is used in this example. The base-m expansion of n:

$$45,113 = 31^3 + 15 \cdot 31^2 + 29 \cdot 31 + 8$$

yields the polynomial $f(x) = x^3 + 15x^2 + 29x + 8$ for which $f(m) \equiv 0 \pmod{n}$ since $f(x)$ was constructed with $f(31) = 45,113$.

The only concern now is that $f(x)$ be irreducible over $\mathbb{Q}$, which amounts to verifying that $f(x)$ does not have any roots over $\mathbb{Q}$ since $f(x)$ is a cubic. The only possible roots are ± 1, ± 2, ± 4, and ± 8. The positive portions of these possibilities can be immediately dispensed with so only four possibilities for roots need be considered. Now $f(-1) = -7$, $f(-2) = 2$, $f(-4) = 68$, and $f(-8) = 224$ so that $f(x)$ has no rational roots and hence is irreducible over $\mathbb{Q}$.

5.2 The Rational Factor Base

The rational factor base consists of prime integers 2, 3, 5, 7, and so on up to a particular bound which is usually determined by experimenting with the smoothness of $a + bm$ for different (a, b) pairs. In this example, all the primes up to 29 are used.

In practice, the rational factor base is stored as pairs $(m \pmod{p}, p)$ for the reasons discussed in §3.8. The only work involved then is computing m modulo various prime integers p up to the desired bound. Table 5.1 details the rational factor base used in this example.

Table 5.1: Rational Factor Base For $n = 45,113$

$(m \pmod{p}, p)$	$(m \pmod{p}, p)$	$(m \pmod{p}, p)$
$(1, 2)$	$(9, 11)$	$(8, 23)$
$(1, 3)$	$(5, 13)$	$(2, 29)$
$(1, 5)$	$(14, 17)$	
$(3, 7)$	$(12, 19)$	

5.3 The Algebraic Factor Base

From §3.1 the algebraic factor base consists of first degree prime ideals of $\mathbb{Z}[\theta]$, which are represented as pairs (r, p) where p is a prime integer and r is a root of $f(x) = x^3 + 15x^2 + 29x + 8$

considered as a polynomial with coefficients in $\mathbb{Z}/p\mathbb{Z}$. Computing the algebraic factor base then amounts to finding roots of $f(x)$ modulo 2, 3, 5, 7 and so on.

Using the methods of §4.2 and the prime 67 as an example, start by computing the polynomial $g(x) = \gcd(f(x), x^{67} - x)$ where $g(x)$ serves to isolate the linear factors of $f(x)$. In this case, $g(x) \equiv f(x) \pmod{67}$ so that $f(x)$ consists of all linear factors and hence must have three roots over $\mathbb{Z}/67\mathbb{Z}$.

Now $g(0) \equiv 8 \pmod{67}$ so that 0 is not a root of $g(x)$. Since $g(x)$ divides $x^{67} - x = x(x^{33}+1)(x^{33}-1)$ it follows that $g(x)$ must divide $(x^{33}+1)(x^{33}-1)$ and in fact

$$g(x) \equiv \gcd(x^{33}+1, g(x)) \cdot \gcd(x^{33}-1, g(x)) \equiv (x^2+21x+21) \cdot (x+61) \pmod{67}. \tag{5.1}$$

Hence, $6 \equiv -61 \pmod{67}$ is a root of $g(x)$ in $\mathbb{Z}/67\mathbb{Z}$ and the pair $(6, 67)$ represents a first degree prime ideal of $\mathbb{Z}[\theta]$ that may be added to the algebraic factor base.

The same process can be used to determine the linear factors of $g_1(x) = x^2 + 21x + 21$. Now $g_1(-1) \equiv 1 \pmod{67}$ so that -1 is not a root of $g_1(x)$ and hence $g_1(x-1)$ must divide $(x^{33}+1)(x^{33}-1)$ for the same reason $g(x)$ does. However, $\gcd(x^{33}-1, g_1(x-1)) \equiv 1 \pmod{67}$ so that $g_1(x-1)$ can't be immediately split into non-trivial factors as was done with $g(x)$ in (5.1).

Continuing on with $g_1(-2) \equiv 50 \pmod{67}$ it is seen that -2 is not a root of $g_1(x)$ and hence $g_1(x-2)$ must then divide $(x^{33}+1)(x^{33}-1)$. This time luck prevails and

$$\begin{aligned} g_1(x-2) &\equiv \gcd(x^{33}+1, g_1(x-2)) \cdot \gcd(x^{33}-1, g_1(x-2)) \\ &\equiv (x+21) \cdot (x+63) \pmod{67}. \end{aligned}$$

The latter yields $46 \equiv -21 \pmod{67}$ and $4 \equiv -63 \pmod{67}$ as roots of $g_1(x-2)$, so that 44 and 2 are roots of $g_1(x)$ over $\mathbb{Z}/67\mathbb{Z}$ and hence the pairs $(2, 67)$ and $(44, 67)$ represent first degree prime ideals of $\mathbb{Z}[\theta]$ which may be used in the algebraic factor base.

Root finding with primes other than 67 is performed in the exact same manner to determine the rest of the algebraic factor base shown in Table 5.2.

5.4 The Quadratic Character Base

Since the quadratic character base of §3.2 is simply a small set of first degree prime ideals of $\mathbb{Z}[\theta]$ that don't occur in the algebraic factor base, in practice one begins searching for roots of $f(x)$ modulo primes q with q strictly greater than the largest prime p occurring in a (r, p) pair in the algebraic factor base. The worked example of §5.3 serves as ample illustration of how the quadratic character base seen in Table 5.3 is computed.

Table 5.2: Algebraic Factor Base For $n = 45,113$

(r, p) pair	(r, p) pair	(r, p) pair	(r, p) pair
$(0, 2)$	$(19, 41)$	$(44, 67)$	$(62, 89)$
$(6, 7)$	$(13, 43)$	$(50, 73)$	$(73, 89)$
$(13, 17)$	$(1, 53)$	$(23, 79)$	$(28, 97)$
$(11, 23)$	$(46, 61)$	$(47, 79)$	$(87, 101)$
$(26, 29)$	$(2, 67)$	$(73, 79)$	$(47, 103)$
$(18, 31)$	$(6, 67)$	$(28, 89)$	

Table 5.3: Quadratic Character Base For $n = 45,113$

(r, p) pair	(r, p) pair	(r, p) pair
$(4, 107)$	$(80, 107)$	$(99, 109)$
$(8, 107)$	$(52, 109)$	

5.5 Sieving

For this example, the sieving interval is chosen such that $-1000 < a < 1000$ for b starting at 1 and proceeding through 2, 3, 4, and so on until more than 39 (a, b) pairs are found with $a + bm$ and $a + b\theta$ smooth. Finding more than 39 pairs will guarantee a linear dependence among the binary vectors associated with those pairs, which leads to perfect squares in $\mathbb{Z}$ and $\mathbb{Z}[\theta]$ as explained in §4.3.

A straight forward implementation technique is to have two sieve arrays in memory, one for $a + bm$ and the other for $\mathrm{N}(a + b\theta)$, each with 2000 entries for all the possible a values for a fixed b. Sieving for smooth values of $a + bm$ proceeds exactly as in §3.5. For instance, the values of a for which $a+bm$ is divisible by the prime 5 for $b = 7$ are of the form $a = -7m+5k$ for $k \in \mathbb{Z}$ such that $-1000 < a < 1000$. From Table 5.1 it is seen that $m \equiv 1 \pmod 5$ and hence a is of the form $a = -7 + 5k$ for $k \in \mathbb{Z}$. The positions in the sieve array for $a + bm$ corresponding to an a value of -997, -992, ... , -12, -7, -2, 3, 8, 13, ... , 993, 998 then have $\ln(5)$ added to their value. This procedure is repeated for all the pairs of Table 5.1. A similar procedure is followed with the (r, p) pairs of Table 5.2 and the sieve array for $\mathrm{N}(a + b\theta)$. Each sieve array is then scanned for positions with positive value in accordance with §3.6 and for such values $a + bm$ and $\mathrm{N}(a + b\theta)$ are trial divided to test for smoothness. The whole procedure is then repeated for the next value of b.

After enough sieving, 40 (a, b) pairs are found with $a + bm$ and $a + b\theta$ smooth, as seen in

Table 5.4: (a,b) Pairs Found During Sieving

(a,b) pair	(a,b) pair	(a,b) pair	(a,b) pair	(a,b) pair	(a,b) pair	(a,b) pair
$(-73,1)$	$(-2,1)$	$(-1,1)$	$(2,1)$	$(3,1)$	$(4,1)$	$(8,1)$
$(13,1)$	$(14,1)$	$(15,1)$	$(32,1)$	$(56,1)$	$(61,1)$	$(104,1)$
$(116,1)$	$(-5,2)$	$(3,2)$	$(25,2)$	$(33,2)$	$(-8,3)$	$(2,3)$
$(17,3)$	$(19,4)$	$(48,5)$	$(54,5)$	$(313,5)$	$(-43,6)$	$(-8,7)$
$(11,7)$	$(38,7)$	$(44,9)$	$(4,11)$	$(119,11)$	$(856,11)$	$(536,15)$
$(5,17)$	$(5,31)$	$(9,32)$	$(-202,43)$	$(24,55)$		

Table 5.4.

5.6 Forming the Matrix

To find the set U in Note 2.4.1, first a binary matrix B is constructed as in §4.3, where a single column of the matrix corresponds to an (a,b) pair found in §5.5 with $a+bm$ and $a+b\theta$ smooth. Since there are 10 primes in the rational factor base, 23 first degree prime ideals in the algebraic factor base, and 5 first degree prime ideals in the quadratic character base, each column of the matrix will have 39 entries, or 39 total rows for the matrix (one bit is added for the sign of $a+bm$).

To show explicitly how the matrix is formed, the column entry for the pair $(-8,3)$ found in §5.5 will be calculated. The first entry is set to 0 since $a+bm = -8+3\cdot 31 = 85$ is positive. The next 10 entries in this column vector are determined from the factorization of $a+bm = 85$ over the rational factor base:

$$85 = 2^0\cdot 3^0\cdot 5^1\cdot 7^0\cdot 11^0\cdot 13^0\cdot 17^1\cdot 19^0\cdot 23^0\cdot 29^0$$

where all the primes in the rational factor base have been shown for clarity. The column vector for $(-8,3)$ then has 10 entries formed by taking the above 10 exponent vectors modulo 2:

$$(0,0,1,0,0,0,1,0,0,0)$$

Next, the norm of $(-8,3)$ is computed and factored over the primes occurring in first degree prime ideal pairs in the algebraic factor base. Recalling from (3.4) that $\mathrm{N}(a+b\theta) = (-b)^d f(-a/b)$ it follows that the norm of an element $a+b\theta$ with $d=3$ and $f(x) =$

$x^3 + 15x^2 + 29x + 8$ can be computed as

$$\begin{aligned}\mathrm{N}(a + b\theta) &= (-b)^3 \cdot \left(\frac{-a^3}{b^3} + 15\frac{a^2}{b^2} - 29\frac{a}{b} + 8 \right) \\ &= a^3 - 15a^2b + 29ab^2 - 8b^3.\end{aligned}$$

The norm of $-8 + 3\theta$ is then $\mathrm{N}(-8 + 3\theta) = -8^3 - 15 \cdot (-8)^2 \cdot 3 - 29 \cdot 8 \cdot 3^2 - 8 \cdot 3^3 = -5696$ and $-5696 = -1 \cdot 2^6 \cdot 89^1$ gives the factorization of that norm over the primes p occurring in the (r, p) pairs of the algebraic factor base.

Note that there can be up to d pairs (r, p) in the algebraic factor base that share the same prime p, but only one such pair can have $a \equiv -br \pmod{p}$. Such an (r, p) pair is the one that will be "responsible" for counting the number of times p divides $\mathrm{N}(a + b\theta)$. In the case for $-8 + 3\theta$, there are three first degree prime ideals in Table 5.2 that have 89 as the prime in their pair representation, specifically $(28, 89)$, $(62, 89)$, and $(73, 89)$. But $-8 \equiv 81 \equiv -3 \cdot 62 \pmod{89}$ so the first degree prime ideal pair $(62, 89)$ is responsible for the exponent of 89. Combining this with the first degree prime ideal having pair $(0, 2)$ yields the next 23 bits in the column vector for (-8, 3):

$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)$$

From §3.2 it follows that to compute a quadratic character for $-8 + 3\theta$ corresponding to the first degree prime ideal represented by the pair (s, q) that the Legendre symbol $\left(\frac{-8+3s}{q}\right)$ must be calculated. Using $(80, 107)$ from the quadratic character base as an example yields

$$\left(\frac{-8 + 3 \cdot 80}{107} \right) = -1.$$

In this case, the vector coordinate for $(80, 107)$ is stored as 1 and would have been stored as 0 had the Legendre symbol been 1, according to §4.3.

Performing the same operations for the remainder of the quadratic character base yields the final 5 bits in the column vector for $(-8, 3)$:

$$(1, 0, 0, 1, 0).$$

The complete 39-bit column vector for $(-8, 3)$ then is seen to be

$$(0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0)^T$$

This same procedure is used on the rest of the (a, b) pairs found in §5.5 to produce the 39×40 binary matrix B.

Table 5.5: (a, b) Pairs Occurring In a Dependency

(a, b) pair	(a, b) pair	(a, b) pair	(a, b) pair	(a, b) pair
$(-1, 1)$	$(104, 1)$	$(-8, 3)$	$(-43, 6)$	$(856, 11)$
$(3, 1)$	$(3, 2)$	$(48, 5)$	$(-8, 7)$	
$(13, 1)$	$(25, 2)$	$(54, 5)$	$(11, 7)$	

5.7 Finding Dependencies

The block Lanczos procedure sketched in §4.5 can be applied to the matrix B constructed in §5.6 to find a dependency among the binary vectors corresponding to the (a, b) pairs of Table 5.4. The resulting (a, b) pairs whose binary vectors occur in a dependency are listed in Table 5.5.

5.8 Computing An Explicit Square Root in $\mathbb{Z}[\theta]$

To reemphasize the statements at the beginning of Chapter 5, it should be noted that the computations performed in this section for determining an explicit value of $\beta \in \mathbb{Z}[\theta]$ for the β of Note 2.4.1 are never actually performed in practice. Since the value $n = 45,113$ in this running example is small, however, the value for β may be computed to check the methods of §5.9, §5.10, and §5.11.

Begin by finding an explicit representation for the product of the $a + b\theta$ values corresponding to the (a, b) pairs found in Table 5.5:

$$\begin{aligned}\delta &= 2051543129764485 \cdot \theta^2 + 15388377355799440 \cdot \theta + 24765692886531904 \\ &= (-1+\theta) \cdot (3+\theta) \cdot (13+\theta) \cdot (104+\theta) \cdot (3+2\theta) \cdot (25+2\theta) \cdot (-8+3\theta) \\ &\quad \cdot (48+5\theta) \cdot (54+5\theta) \cdot (-43+6\theta) \cdot (-8+7\theta) \cdot (11+7\theta) \cdot (856+11\theta)\end{aligned} \tag{5.2}$$

where all the computations in (5.2) are treated as multiplication of polynomials modulo $f(x) = x^3 + 15x^2 + 29x + 8$ with θ substituted for x. Next, the value $f'(\theta)^2 = 138 \cdot \theta^2 + 363 \cdot \theta + 481 \in \mathbb{Z}[\theta]$ is computed and multiplied by (5.2) to yield

$$\begin{aligned}f'(\theta)^2 \cdot \delta = 22455983949710645412 \cdot \theta^2 \\ + 54100105785512562427 \cdot \theta + 22939402657683071224\end{aligned}$$

which is the square of an element $\beta \in \mathbb{Z}[\theta]$. Indeed, without too much more effort it is seen that

$$\beta = 599923511 \cdot \theta^2 + 3686043120 \cdot \theta + 3889976768$$

satisfies $\beta^2 = f'(\theta)^2 \cdot \delta$ in $\mathbb{Z}[\theta]$.

Applying the ring homomorphism $\phi : \mathbb{Z}[\theta] \to \mathbb{Z}/45113\mathbb{Z}$ of Proposition 2.4.1 to β gives

$$\begin{aligned} x = 43922 \equiv 694683807559 \equiv \\ 599923511 \cdot 31^2 + 3686043120 \cdot 31 + 3889976768 \equiv \phi(\beta) \pmod{45331} \end{aligned}$$

and it is this value $x = 43,922$ that should result from the square root techniques of §5.9, §5.10, and §5.11.

5.9 Determining Applicable Finite Fields

The first stage in computing $x = \phi(\beta) \pmod{p}$ using the methods outlined in §4.6, §4.7, and §4.8 is to find a number of finite fields that are "compatible" with $\mathbb{Q}(\theta)$, which boils down to finding prime integers p for which $f(x)$ remains irreducible modulo p. First, $f(x) = x^3 + 15x^2 + 29x + 8$ will be tested for irreducibility modulo the prime $p = 9929$ using the techniques of §4.9.

Begin by computing $x^p - x$ modulo $f(x)$

$$x^{9929} - x \equiv 7449x^2 + 4697x + 5984 \pmod{f(x)}$$

and then taking the greatest common divisor with $f(x)$ gives

$$\gcd(7449x^2 + 4697x + 5984, x^3 + 15x^2 + 29x + 8) = 1$$

when considered modulo 9929. Thus, $f(x)$ is irreducible over $\mathbb{Z}/9929\mathbb{Z}$ by Theorem 4.9.5 and the finite field $\mathbb{F}_{9929^3}$ may be used in the methods of §4.6, §4.7, and §4.8.

As an example of a value for p for which $f(x)$ does not turn out to be irreducible, if $p = 9923$ then

$$x^{9923} - x \equiv 7726x^2 + 1477x + 7301 \pmod{f(x)}$$

and the greatest common divisor turns out to be

$$\gcd(7726x^2 + 1477x + 7301, x^3 + 15x^2 + 29x + 8) = x - 847$$

modulo 9923. Hence 847 is a root of $f(x)$ modulo 9923 and therefore $f(x)$ is not irreducible over $\mathbb{Z}/9923\mathbb{Z}$.

Table 5.6 lists the three primes p_i for which the finite field $\mathbb{F}_{p_i^3}$ will be used in §5.10 and §5.11 for computing x in §5.8.

Table 5.6: Primes Determining Finite Fields $\mathbb{F}_{p_i^3}$

p_0	p_1	p_2
9851	9907	9929

5.10 Square Roots in a Finite Field

The techniques of §4.8 will be illustrated for the prime $p = 9929$ and the finite field $\mathbb{F}_{p^3}$, where the elements of the latter field may be represented as polynomials in θ_p where θ_p is a root of $f(x)$ in the splitting field of $f(x) = x^3 + 15x^2 + 29x + 8$ over $\mathbb{Z}/p\mathbb{Z}$ by Theorem 4.7.2. Denote $\mathbb{F}_{p^3}$ by $\mathbb{F}_p(\theta_p)$ as well.

Begin by letting $q = p^3$ so that $q - 1 = 2^r \cdot s$ where $r = 3$ and $s = 122356359011$. The first task is to find a quadratic non-residue in $\mathbb{F}_p(\theta_p)$, which is very easy since exactly half the elements satisfy this property. A direct search immediately yields $\theta_p + 1$ as a non-residue by Theorem 4.8.1 since

$$(\theta_p + 1)^{\frac{9929^3 - 1}{2}} \equiv -1 \pmod{9929}.$$

From Theorem 4.8.3 it follows that the Sylow 2-subgroup S_8 of $\mathbb{F}_p(\theta_p)$ can then be represented by the set

$$S_8 = \left\{1, (\theta_p + 1)^s, (\theta_p + 1)^{2s}, (\theta_p + 1)^{3s}, \ldots, (\theta_p + 1)^{7s}\right\}.$$

Next, the element δ of §5.8 is computed in $\mathbb{F}_p(\theta_p)$ as

$$\begin{aligned}\delta_p = 2027\theta_p^2 + 3891\theta_p + 6659 \equiv f'(\theta_p)^2 \cdot (-1 + \theta_p) \cdot (3 + \theta_p) \cdot (13 + \theta_p) \cdot \\ (104 + \theta_p) \cdot (3 + 2\theta_p) \cdot (25 + 2\theta_p) \cdot (-8 + 3\theta_p) \cdot (48 + 5\theta_p) \cdot \\ (54 + 5\theta_p) \cdot (-43 + 6\theta_p) \cdot (-8 + 7\theta_p) \cdot (11 + 7\theta_p) \cdot (856 + 11\theta_p)\end{aligned}$$

The immediate goal is then to find an element $\zeta \in S_8$ with $\zeta^2 \equiv \delta^s$. By direct computation in $\mathbb{F}_p(\theta_p)$ it is seen that $\delta^s \equiv 9928 \pmod{9929}$ and from Table 5.7 it follows that $\zeta = (\theta_p + 1)^{2s} \equiv 2102 \pmod{9929}$ is an element with the desired property.

If $\omega = \delta^{(s+1)/2}$ then $\nu = \omega \cdot \zeta^{-1}$ is a square root of δ since

$$\nu^2 \equiv \omega^2 \cdot \zeta^{-2} \equiv \delta^s \cdot \delta \cdot \delta^{-s} \equiv \delta$$

since ζ was found such that $\zeta^2 \equiv \delta^s$. Computing out ν explicitly involves computing the multiplicative inverse 7827 of $\zeta = 2102$ modulo 9929 and then multiplying out

$$\nu = \omega \cdot \zeta^{-1} \equiv 7827 \cdot \delta^{\frac{s+1}{2}} \equiv 3402\theta_p^2 + 1160\theta_p + 3077.$$

Table 5.7: Members of S_8 and Their Squares

i	$(\theta_p+1)^{is}$	$(\theta_p+1)^{2is}$	i	$(\theta_p+1)^{is}$	$(\theta_p+1)^{2is}$
0	1	1	4	9928	1
1	1273	2102	5	8656	2102
2	2102	9928	6	7827	9928
3	4945	7827	7	4984	7827

Note that the other square root of δ is simply $-\nu$ in $\mathbb{F}_p(\theta_p)$, which is seen to be $6527\theta_p^2 + 8769\theta_p + 6852$

In the above description, the element ζ was found from the list of all its possible values in Table 5.7. The actual square root algorithm employed in §4.8 uses an iterative approach to avoid the explicit computation of all elements in S_8. First, λ_0 is initialized as $\lambda_0 = -1 \equiv \delta^s \pmod{9929}$ and ω_0 is initialized as $\omega_0 = 2124\theta_p^2 + 5175\theta_p + 4075 \equiv \delta^{(s+1)/2}$. Finally, the variable m is set to 1 since the order of λ_0 in $\mathbb{F}_{p^3}^*$ is seen to be 2.

The iterative process begins by calculating λ_1 and ω_1 from λ_0 and ω_0. Letting $\zeta = (\theta_p+1)^s \equiv 1273 \pmod{9929}$ then

$$\lambda_1 \equiv \lambda_0\zeta^{2^{r-m}} \equiv -1 \cdot 1273^{2^{3-1}} \equiv -1 \cdot 1273^4 \equiv 1 \pmod{9929}$$

and

$$\omega_1 = \omega_0\zeta^{2^{r-m-1}} \equiv \omega_0 \cdot 1273^{2^{3-1-1}} \equiv \omega_0 \cdot 1273^2 \equiv 6527\theta_p^2 + 8769\theta_p + 6852.$$

Since $\lambda_1 = 1$ it follows that ω_1 is a square root of δ, which is verified from the earlier calculation using Table 5.7.

5.11 Using the Chinese Remainder Theorem

From §5.8 it is known that the value of x is 694683807559 and hence $x \pmod{45113} = 43922$. The goal of this section is to show that this value can be computed once square roots of δ from §5.8 in the finite fields of Table 5.6 are known, as is done in §5.10. The relevant values from the computation of square roots in the finite fields of Table 5.6 are summarized in Table 5.8.

For the sake of explanation, the value of z from §4.6 is computed as

$$z = \sum_{i=0}^{2} a_i x_i P_i = 7261482164111988.$$

Table 5.8: Square Roots of δ in Finite Fields

	$i = 0$	$i = 1$	$i = 2$
root	$7462 \cdot \theta_{p_i}^2 +$ $5791 \cdot \theta_{p_i} + 4037$	$5126 \cdot \theta_{p_i}^2 +$ $5072 \cdot \theta_{p_i} + 3125$	$3402 \cdot \theta_{p_i}^2 +$ $1160 \cdot \theta_{p_i} + 3077$
x_i	5694	4152	2002
a_i	7174	3691	8928
p_i	9851	9907	9929
P_i	98366603	97810579	97593857

The value of P can also be computed as $P = 9851 \cdot 9907 \cdot 9929 = 969009406153$. Then it is easily verified that $z \equiv x \pmod{P}$ and in fact $694683807559 = 7261482164111988 - 7493 \cdot 969009406153$ gives the value $r = 7493$ in $x = z - rP$. Note that these computations are never carried out in practice and are done here only to verify the results of the remaining portions of the example. In the rest of this section it should be noted that all calculations take place modulo n, which avoids the problems of unbounded integers when using direct computations.

Begin by computing the value of r using

$$\frac{z}{P} = \sum_{i=0}^{2} \frac{a_i x_i}{p_i} = \frac{7174 \cdot 5694}{9851} + \frac{3691 \cdot 4152}{9907} + \frac{8928 \cdot 2002}{9929} = 7493.72.$$

Rounding the last result yields $r = 7493$, which is indeed the correct value for r found by direct calculation. In this computation it should be noted that neither x nor z needs to be calculated explicitly; only relatively small floating point numbers are required.

The value for $x \pmod{n}$ can now be computed as

$$\begin{aligned}
x \pmod{n} &\equiv \sum_{i=0}^{2} a_i x_i P_i \pmod{n} - rP \pmod{n} \\
&\equiv (a_0 x_0 p_1 p_2) \pmod{n} + (a_1 x_1 p_0 p_2) \pmod{n} \\
&\quad + (a_2 x_2 p_0 p_1) \pmod{n} - (r p_0 p_1 p_2) \pmod{n} \\
&\equiv 7174 \cdot 5694 \cdot 9907 \cdot 9929 \pmod{45113} \\
&\quad + 3691 \cdot 4152 \cdot 9851 \cdot 9929 \pmod{45113} \\
&\quad + 8928 \cdot 2002 \cdot 9851 \cdot 9907 \pmod{45113} \\
&\quad - 7493 \cdot 9851 \cdot 9907 \cdot 9929 \pmod{45113} \\
&\equiv (41457 + 26833 + 42022 - 21277) \pmod{45113} \\
&\equiv 43922 \pmod{45113}
\end{aligned}$$

and the result is achieved. Again it it stressed that at no point in the above computation are explicit values for x or z needed, and no intermediate result ever exceeds the size of n.

5.12 Putting It All Together

Multiplying out the values of $a + bm$ for the (a, b) pairs of Table 5.5 yields the following square in $\mathbb{Z}$:

$$\begin{aligned}31746503388600^2 = 1007840477402391282609960000 = \\ (-1+31)\cdot(3+31)\cdot(13+31)\cdot(104+31)\cdot(3+2\cdot 31)\cdot \\ (25+2\cdot 31)\cdot(-8+3\cdot 31)\cdot(48+5\cdot 31)\cdot(54+5\cdot 31)\cdot \\ (-43+6\cdot 31)\cdot(-8+7\cdot 31)\cdot(11+7\cdot 31)\cdot(856+11\cdot 31)\end{aligned} \tag{5.3}$$

From Note 2.4.1 it follows that (5.3) must also be multiplied by $f'(m)^2 = (3\cdot 31^2 + 30\cdot 31 + 29)^2 = 3824^2$. Then set

$$y = 15160 \equiv 3824 \cdot 31746503388600 \pmod{45113}.$$

From the calculations in §5.11 it was found that $x = 43922 \pmod{45113}$, and furthermore x^2 and y^2 are equivalent modulo n by the ring epimorphism ϕ of Proposition 2.4.1. Thus

$$15160^2 \equiv 43922^2 \pmod{45113}.$$

Then $45,113$ divides $15160^2 - 43922^2 = (15160-43922)\cdot(15160+43922)$ and this difference of squares in turn yields a non-trivial factorization of $45,113$ since $\gcd(15160-43922, 45113) = 197$ and $\gcd(15160+43922, 45113) = 229$. Note that these values for x and y yield a non-trivial factorization since each of $x - y$ and $x + y$ contains exactly one non-trivial factor of n:

$$43922 - 15160 = 2\cdot 73\cdot 197 \quad \text{and} \quad 43922 + 15160 = 2\cdot 3\cdot 43\cdot 229.$$

Chapter 6

Polynomial Selection and Parameter Tuning

The first step of GNFS, selecting a polynomial, also happens to also be most important step. For any given integer n that is to be factored, there can literally be billions of choices for the polynomial $f(x)$ and the integer m with $f(m) \equiv 0 \pmod{n}$. From real-world experiments with the GNFS it has been seen that in many cases some polynomials are remarkably better than others, where a polynomial $f(x)$ is considered "better" than another polynomial $g(x)$ if more (a, b) pairs with $a + bm$ and $\langle a + b\theta \rangle$ smooth are found with $f(x)$ than with $g(x)$ using the same sieve interval, factor base sizes, and quadratic character base sizes. Although a crucial stage of the GNFS, polynomial selection still remains an *ad-hoc* and underdeveloped area. Success relies upon some basic heuristic reasoning, lots of experimentation, and often times just blind luck.

The goal here, then, is to explain how the experimentation phase of the GNFS proceeds, starting with an initial n to factor, proceeding on to trials of various candidate polynomials, and concluding with tuning the sieve for a particular selection of polynomial, factor base, and sieve interval.

6.1 Tweaking the Base-m Method

One begins polynomial selection by choosing an integer m close to $n^{1/d}$ such that

$$n = c_d \cdot m^d + c_{d-1} \cdot m^{d-1} + \cdots + c_1 \cdot m + c_0$$

and $c_d = 1$. Then $f(x)$ is defined to be $f(x) = x^d + c_{d-1}x^{d-1} + \cdots + c_1 x + c_0$ and by construction then $f(m) = n \equiv 0 \pmod{n}$. If $f(x)$ turns out to be reducible, then a non-trivial factorization of n is likely to be available immediately, in which case the process terminates successfully.

One prominent strategy for producing "better" polynomials has been to reduce the sizes of the coefficients c_i in $f(x)$, with the idea being that the smaller these coefficients are, the more likely $\mathrm{N}(a + b\theta)$ is to be small. If $\mathrm{N}(a + b\theta)$ is relatively small it stands to reason that it is more likely to factor over a small set of primes, and hence the more likely $\langle a + b\theta \rangle$ is to factor over the algebraic factor base.

With this in mind, begin by noting that the base-m method produces an $f(x)$ such that $|c_i| < m$ for $0 \leq i < d$. It turns out that this can be improved to $|c_i| < m/2$, as follows. For any $c_i > m/2$ then $m - c_i < m/2$. If $0 \leq i < d - 1$ then

$$(c_{i+1} + 1) \cdot m^{i+1} - (m - c_i) \cdot m^i = c_{i+1} \cdot m^{i+1} + m^{i+1} - m^{i+1} + c_i \cdot m_i = c_{i+1} \cdot m^{i+1} + c_i m_i.$$

Hence, one can obtain an equivalent base-m expansion for n by replacing c_{i+1} with $c_{i+1} + 1$ and c_i with $-(m - c_i)$ for any $c_i > m/2$. This expansion has the benefit that all coefficients then have absolute value less than $m/2$, if negative coefficients are allowed. Note i is chosen less than $d - 1$ so that the polynomial stays monic.

6.2 Using Polynomials of the Form$f(x) + g(x)$

The technique outlined in §6.1 of adjusting an existing polynomial $f(x)$ with $f(m) \equiv 0 \pmod{n}$ can be expanded upon by adding another polynomial $g(x)$ for which $g(m) = 0$. Such polynomials are not too difficult to come up with, and in fact if

$$g(x) = \sum_{i=1}^{d-1} c_i \cdot (x^i - mx^{i-1}) \tag{6.1}$$

then it is easily seen that $g(m) = 0$ since

$$g(m) = \sum_{i=1}^{d-1} c_i \cdot (m^i - m \cdot m^{i-1}) = \sum_{i=1}^{d-1} c_i \cdot (m^i - m^i) = 0.$$

To summarize, polynomial selection begins with an integer $m \approx n^{1/d}$ and proceeds by expanding n base-m. The polynomial generated by this method can be further "tweaked" to insure that all coefficients have absolute value less than $m/2$. With or without this adjustment, the polynomials of the base-m method can lead to other polynomials by adding polynomials of the form shown in (6.1). Note polynomials of the form seen in (6.1) are produced simply by choosing different values for $c_1, c_2, \ldots, c_{d-1}$.

6.3 Finding a Good Polynomial

Having chosen candidate polynomials for GNFS, the next step is to choose a few different sizes for factor bases and sieve intervals, and then simply to sieve each polynomial using these different parameter choices. For the sake of brevity, the term “smooth” is applied here to an (a, b) pair if that pair has $a + bm$ and $\langle a + b\theta\rangle$ smooth over their respective factor bases. The idea here is to find a polynomial that produces the most smooth $(a + b)$ pairs in the shortest amount of time. For any given polynomial and factor base, the sieve interval is varied and for each interval the number of smooth values found and the time it took to find them is recorded. An estimate for the ratio of time taken to smooth (a, b) pairs can then be made and compared with other ratios for the same polynomial and factor base. The interval with the smallest ratio is then selected and multiplied by an estimate for the total number of pairs needed (rational primes + algebraic primes + quadratic characters will suffice) to estimate the shortest amount of time it will take to produce enough pairs, for the given polynomial and factor base. This latter statistic is recorded, and then the same procedure for different factor bases with the same polynomial is performed. Once “enough” factor bases have been examined, the factor base with the shortest time estimate for producing (a, b) pairs is chosen. This last figure represents the best performance for a particular polynomial, over a variety of factor bases and sieve intervals. The whole procedure can then be done again for other polynomials, with the polynomial being selected for the GNFS being the one with the shortest estimated time to produce enough (a, b) pairs.

6.4 Example Polynomial Selection

As an example of the different polynomial selection procedures and the aforementioned strategy for selecting factor bases and sieve intervals, these techniques will be applied to the number

$$n = 556158012756522140970101270050308458769458529626977$$

where n is seen to be a 51 digit integer. Polynomials of degree $d = 3$ are generally used in the GNFS for numbers of this size.

Begin by noting that

$$n^{1/3} \approx 82236774153802891$$

so set $m = 82236774153802891$ and perform a base-m expansion on n to get

$$\begin{aligned} n = (82236774153802891)^3 + 27709956990112403 \cdot \\ (82236774153802891) + 45334438077235933 \end{aligned}$$

and hence $f_1(x) = x^3 + 27709956990112403x + 45334438077235933$ is a candidate polynomial for n.

One can adjust the polynomial $f_1(x)$ by adding a polynomial $g_1(x)$ of the form $g_1(x) = c1 \cdot (x - m) + c2 \cdot (x^2 - m \cdot x)$, where $c1$ and $c2$ are usually chosen small to keep the coefficients of $g_1(x)$ small. Letting $c1 = 1$ and $c2 = -1$ gives $g_1(x) = -x^2 + 82236774153802892x - 82236774153802891$ and a new candidate polynomial $f_2(x)$ is produced as

$$f_2(x) = f_1(x) + g_1(x) = x^3 - x^2 + 109946731143915295x - 36902336076566958.$$

Another candidate $f_3(x)$ can be produced simply by altering the value of m by a "small" amount and re-running the base-m algorithm with this new value of m. In fact, the value of m can vary a great deal because of the magnitude of the integers involved. Subtracting 107 from the original value for m used above yields a new value of $m = 82236774153802784$, and the base-m method applied here produces

$$f_3(x) = x^3 + 321x^2 + 27709956990146786x + 49775966483587873.$$

In this case, $m/2 = 41118387076901392$ and the constant term coefficient in $f_3(x)$ happens to be larger than this value. One can then form a new polynomial $f_4(x)$ from $f_3(x)$ by replacing the constant term coefficient of $f_3(x)$ with the value of m subtracted from it, and incrementing the monic term coefficient by one, yielding

$$f_4(x) = x^3 + 321x^2 + 27709956990146787x - 32460807670214911.$$

Note that for all these polynomials, $f_i(m) \equiv 0 \pmod{n}$ and $f_i(x)$ is irreducible. The next stage is to see which of the polynomials can produce the most smooth (a, b) pairs in the shortest amount of time.

6.5 Testing the Example Polynomials

Using the ideas outlined in §6.3 each polynomial has a sieve run with three different sizes for factor bases, and each of these choices is sieved over five different sieve intervals. The relevant performance data is summarized in Appendix A.

Begin the analysis of the performance data by using Table A.12 as an example. Note that as the length of the sieve interval increases the number of smooth (a, b) pairs found also increases. This seems natural, as one would expect the number of smooth pairs to increase as the number of pairs examined increases. It should also be noted that the amount of time required for finding these pairs also increases as the interval size increases, so steadily in fact that it seems the best policy is to stay with the smallest sieving interval. At this point, it should be noted that smooth (a, b) pairs are difficult to find, and as shown in practical experience, the number of smooth pairs steadily decreases as a and b increase. Thus, it is sound advice to choose a sieving interval that yields a healthy number of smooth pairs, without compromising speed too much. If one uses the timing statistics for the third sieving interval to estimate the length of time required to find enough smooth (a, b) pairs, it works out to

$$0.89055 \times 22,175 \text{ seconds} \approx 5.5 \text{ hours},$$

which is only 48 minutes more expensive than the shortest sieving interval. In practice, since the third sieving interval produced over one and a half times as many smooth (a, b) pairs as the shortest sieving interval, it is probably the best one to use for this choice of polynomial and factor base.

After examining the data, the polynomial $f_4(x)$ stands out. Its performance ratio is better than all other polynomials for almost every choice of factor base sizes and sieve interval.

6.6 The Guessing Game

In should be noted how unpredictable the number of smooth (a, b) pairs produced is for any given selection of a polynomial, factor base, and sieving interval, even with this small data set. For example, it appears as if the time required for finding smooth pairs increases as the sieve interval increases for all the data examined, however that is not the case. There is actually a slight decrease in time when moving from the shortest interval to the next shortest interval in Table A.1. However, that is not the case for the same polynomial and the same intervals with larger factor bases, as seen in Table A.2 and Table A.3. Thus, a sieve interval may be extremely good for a polynomial with one factor base but produce terrible results for the same polynomial with a different factor base. Another trend that seems evident is that increasing the sizes of the factor bases seems to decrease the overall estimated time for finding enough smooth (a, b) pairs. However as the factor base sizes increase this will not always be true, since the larger primes in the larger factor bases occur in fewer $a + bm$ and $\mathrm{N}(a + b\theta)$ and hence are sieved with unnecessarily. Something of this sort is evidenced in Table A.4 and Table A.5, where unlike any of the other polynomials, the estimated time increases from the smaller factor base to the larger. The inexplicable part comes from Table A.6, where the estimated time actually *decreases* from Table A.5.

To further debunk any kind of first glance intuition, one might initially guess that the polynomial $f_1(x)$ would produce the most smooth pairs, since it has a coefficient of 0 for its quadratic term. Indeed, it is the second-best performing polynomial, but as seen in the data, the best polynomial $f_4(x)$ also has the largest quadratic coefficient of all the polynomials tested. On the other hand, the worst-performing polynomial $f_3(x)$ has the same value as $f_4(x)$ for its quadratic term.

Even with a very small set of test cases, one can see that polynomial selection is still very much a “black” art that defies any sort of common sense intuition. There are billions of candidate polynomials for large integers n and potentially a great deal of unexploited structure relating these polynomials and their probability of smoothness that needs to be investigated.

6.7 An Alternate Strategy

A variation on the strategy detailed above cuts down dramatically on the amount of work that goes into screening candidate polynomials, with the end result being that more candidate polynomials can be examined in a shorter amount of time. Upon examining the performance data in Appendix A it is generally seen that when one polynomial performs better than another for particular choices of factor base sizes and sieve interval, that polynomial will perform better than the other for all choices of factor base sizes and sieve intervals. The basic idea then is to judge all polynomials against one choice of factor base sizes and sieve interval, stopping when a polynomial with dramatically better performance than any other is found. That polynomial is then further optimized by running sieves with different factor base sizes and sieve intervals as in §6.3. By using only a single selection of factor base sizes and sieve interval, the number of polynomials that can be examined in a given amount of time is greatly increased.

More specifically, choose a polynomial and run multiple tests of factor base sizes and sieve intervals, as detailed in §6.3. When a mid-range parameter choice that balances the number of smooth (a, b) pairs produced with the time required is found, that choice of factor base sizes and sieve interval will be be used as the test parameters for the remaining candidate polynomials. All polynomials then run through a single test with the same factor base sizes and sieve interval, and the one with the least expected run time to find enough smooth pairs is kept. The original test with multiple factor base sizes and sieve intervals can then be run on the selected polynomial to further hone its performance. In fact, once the selection of factor base sizes and sieve interval are made, an automated tool can generate polynomials by varying values for m, adjusting the coefficients of the base-m method, and substituting small values for c_1, c_2, ... , c_{d-1}. The tool can further keep performance data while sieving with the polynomials it generated. An end-user can then analyze the performance data for the various candidate polynomials when the tool is finished and choose the one that performs best.

Appendix A

Example Performance Data

Table A.1: $f_1(x)$ with small factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	10,000	50	14,253
Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	30	1 min 16 sec	2.53333 sec/pair
400,000	38	1 min 36 sec	2.52632 sec/pair
500,000	47	2 min 24 sec	3.06383 sec/pair
1,000,000	88	4 min 3 sec	2.76136 sec/pair
1,500,000	124	7 min 22 sec	3.56452 sec/pair
Using best ratio would take $2.52632 \times 14,253$ sec ≈ 10 hours to get enough relations			

Table A.2: $f_1(x)$ with medium factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	14,286	50	18,539
Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	54	1 min 23 sec	1.53704 sec/pair
400,000	65	1 min 57 sec	1.80000 sec/pair
500,000	79	2 min 32 sec	1.92405 sec/pair
1,000,000	139	4 min 50 sec	2.08633 sec/pair
1,500,000	202	7 min 2 sec	2.08911 sec/pair
Using best ratio would take $1.53704 \times 18,539$ sec ≈ 8 hours to get enough relations			

Table A.3: $f_1(x)$ with large factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
7,837	14,286	50	22,173

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	102	1 min 28 sec	0.86275 sec/pair
400,000	127	1 min 50 sec	0.86614 sec/pair
500,000	153	2 min 38 sec	1.03268 sec/pair
1,000,000	249	5 min 10 sec	1.24498 sec/pair
1,500,000	336	7 min 59 sec	1.42560 sec/pair

Using best ratio would take $0.86275 \times 22,173$ sec ≈ 5.3 hours to get enough relations

Table A.4: $f_2(x)$ with small factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	10,132	50	14,385

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	45	1 min 26 sec	1.91111 sec/pair
400,000	54	2 min 1 sec	2.24074 sec/pair
500,000	65	2 min 32 sec	2.33846 sec/pair
1,000,000	106	5 min 11 sec	2.93396 sec/pair
1,500,000	141	7 min 32 sec	3.20567 sec/pair

Using best ratio would take $1.91111 \times 14,385$ sec ≈ 7.6 hours to get enough relations

Table A.5: $f_2(x)$ with medium factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	14,296	52	18,551

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	59	1 min 31 sec	1.54237 sec/pair
400,000	75	2 min 6 sec	1.68000 sec/pair
500,000	90	2 min 43 sec	1.81111 sec/pair
1,000,000	147	5 min 29 sec	2.23810 sec/pair
1,500,000	198	7 min 53 sec	2.38889 sec/pair

Using best ratio would take $1.54237 \times 18,551$ sec ≈ 7.9 hours to get enough relations

Table A.6: $f_2(x)$ with large factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
7,837	14,296	52	22,185

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	100	1 min 37 sec	0.97000 sec/pair
400,000	124	2 min 15 sec	1.08871 sec/pair
500,000	149	2 min 52 sec	1.15436 sec/pair
1,000,000	246	5 min 45 sec	1.40244 sec/pair
1,500,000	338	8 min 26 sec	1.49704 sec/pair

Using best ratio would take $0.97 \times 22,185$ sec $\approx$ 6 hours to get enough relations

Table A.7: $f_3(x)$ with small factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	10,037	50	14,290

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	27	1 min 18 sec	2.88889 sec/pair
400,000	37	1 min 51 sec	3.00000 sec/pair
500,000	44	2 min 22 sec	3.22727 sec/pair
1,000,000	74	4 min 35 sec	3.71622 sec/pair
1,500,000	100	6 min 39 sec	3.99000 sec/pair

Using best ratio would take $2.88889 \times 14,290$ sec $\approx$ 11.5 hours to get enough relations

Table A.8: $f_3(x)$ with medium factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	14,296	50	18,550

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	41	1 min 22 sec	2.14634 sec/pair
400,000	57	1 min 57 sec	2.05263 sec/pair
500,000	69	2 min 28 sec	2.14493 sec/pair
1,000,000	118	4 min 37 sec	2.34746 sec/pair
1,500,000	161	7 min 4 sec	2.63354 sec/pair

Using best ratio would take $2.05263 \times 18,550$ sec $\approx$ 10.6 hours to get enough relations

Table A.9: $f_3(x)$ with large factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
7,837	14,296	50	22,184

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	87	1 min 30 sec	1.03448 sec/pair
400,000	115	2 min 7 sec	1.10435 sec/pair
500,000	136	2 min 40 sec	1.17647 sec/pair
1,000,000	220	5 min 23 sec	1.46818 sec/pair
1,500,000	298	7 min 51 sec	1.58054 sec/pair

Using best ratio would take 1.03448 × 22,184 sec ≈ 6.4 hours to get enough relations

Table A.10: $f_4(x)$ with small factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	10,113	50	14,366

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	50	1 min 29 sec	1.78000 sec/pair
400,000	64	2 min 4 sec	1.93750 sec/pair
500,000	74	2 min 38 sec	2.13514 sec/pair
1,000,000	109	5 min 18 sec	2.91743 sec/pair
1,500,000	145	7 min 46 sec	3.21379 sec/pair

Using best ratio would take 1.78 × 14,366 sec ≈ 7.1 hours to get enough relations

Table A.11: $f_4(x)$ with medium factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
4,203	14,288	50	18,541

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	83	1 min 34 sec	1.13253 sec/pair
400,000	102	2 min 12 sec	1.29412 sec/pair
500,000	117	2 min 44 sec	1.40171 sec/pair
1,000,000	187	5 min 33 sec	1.78075 sec/pair
1,500,000	241	8 min 7 sec	2.02075 sec/pair

Using best ratio would take 1.13253 × 18,541 sec ≈ 5.8 hours to get enough relations

Table A.12: $f_4(x)$ with large factor base

Rational Primes	Algebraic Primes	Quadratic Characters	Total
7,837	14,288	50	22,175

Sieve Interval	Pairs Found	Total Time	Time per Pair
300,000	130	1 min 40 sec	0.76923 sec/pair
400,000	168	2 min 21 sec	0.83929 sec/pair
500,000	201	2 min 59 sec	0.89055 sec/pair
1,000,000	320	5 min 56 sec	1.11250 sec/pair
1,500,000	428	8 min 43 sec	1.22196 sec/pair

Using best ratio would take $0.76923 \times 22,175$ sec ≈ 4.7 hours to get enough relations

Bibliography

[1] Leonard M. Adleman, *Factoring numbers using singular integers*, Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing (STOC) (New Orleans), 1991, pp. 64–71.

[2] David M. Bressoud, *Factorization and primality testing*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1989.

[3] John Brillhart, D.H. Lehmer, J.L. Selfridge, Bryant Tuckerman, and S.S. Wagstaff, Jr., *Factorizations of* $b^n \pm 1$*,* $b = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers*, second ed., Contemporary Mathematics, vol. 22, American Mathematical Society, Providence, Rhode Island, 1988.

[4] J. Buchmann, J.Loho, and J.Zayer, *An implementation of the general number field sieve*, Advances in Cryptology (Proceedings of Crypto '93) (Berlin), Lecture Notes in Computer Science, no. 773, Springer-Verlag, 1994, pp. 159–165.

[5] J.P. Buhler, H.W. Lenstra, Jr., and Carl Pomerance, *Factoring integers with the number field sieve*, In Lenstra and Lenstra [17], pp. 50–94.

[6] Henri Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics, vol. 138, Springer-Verlag, Berlin, 1993.

[7] Jean-Marc Couveignes, *Computing a square root for the number field sieve*, In Lenstra and Lenstra [17], pp. 95–102.

[8] A.K. Lenstra Daniel J. Bernstein, *A general number field sieve implementation*, In Lenstra and Lenstra [17], pp. 103–126.

[9] Bruce Dodson and Arjen K. Lenstra, *NFS with four large primes: an explosive experiment*, Advances in Cryptology (Crypto '95) (Berlin), Lecture Notes in Computer Science, no. 963, Springer-Verlag, 1995, pp. 372–385.

[10] David S. Dummit and Richard M. Foote, *Abstract algebra*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[11] Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence, *Linear algebra*, second ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[12] R.M. Huizing, *An implementation of the number field sieve*, Experimental Mathematics **5** (1996), no. 3, 231–253.

[13] ______, *A multiple polynomial general number field sieve*, Algorithmic Number Theory (Berlin), Lecture Notes in Computer Science, vol. 1122, Springer-Verlag, 1996, pp. 99–114.

[14] Thomas W. Hungerford, *Algebra*, Graduate Texts in Mathematics, vol. 73, Springer-Verlag, New York, 1974.

[15] Donald E. Knuth, *Seminumerical algorithms*, second ed., The Art of Computer Programming, vol. 2, Addison-Wesley, Reading, Massachusetts, 1981.

[16] Ramanujachary Kumanduri, *Number theory with computer applications*, Prentice-Hall, Upper Saddle River, New Jersey, 1998.

[17] A.K. Lenstra and H.W. Lenstra, Jr. (eds.), *The development of the number field sieve*, Lecture Notes in Mathematics, vol. 1554, Berlin, Springer-Verlag, 1993.

[18] A.K. Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard, *The factorization of the ninth Fermat number*, Mathematics of Computation **61** (1993), no. 203, 319–349.

[19] ______, *The number field sieve*, In Lenstra and Lenstra [17], pp. 11–42.

[20] A.K. Lenstra and M.S. Manasse, *Factoring with two large primes*, Mathematics of Computation **63** (1994), no. 208, 785–798.

[21] Rudolf Lidl and Harald Niederreiter, *Introduction to finite fields and their applications*, revised ed., Cambridge University Press, Cambridge, 1994.

[22] Peter L. Montgomery, *Square roots of products of algebraic numbers*, Proceedings of Symposia in Applied Mathematics, Mathematics of Computation 1943–1993 (Vancouver) (Walter Gautschi, ed.), 1993.

[23] ______, *A block Lanczos algorithm for finding dependencies over GF(2)*, Advances in Cryptology – EUROCRYPT '95 (Berlin) (L.C. Guillou and J.J. Quisquater, eds.), Lecture Notes in Computer Science, vol. 921, Springer-Verlag, 1995, pp. 106–120.

[24] Ivan Niven, Herbert S. Zuckerman, and Hugh L. Montgomery, *An introduction to the theory of numbers*, fifth ed., John Wiley and Sons, New York, 1991.

[25] J.M. Pollard, *The lattice sieve*, In Lenstra and Lenstra [17], pp. 43–49.

[26] Carl Pomerance, *A tale of two sieves*, Notices of the American Mathematical Society **43** (1996), no. 12, 1473–1485.

[27] Hans Riesel, *Prime numbers and computer methods of factorization*, second ed., Progress in Mathematics, vol. 126, Birkhäuser, Boston, 1994.

[28] Kevin S. McCurley Roger A. Golliver, Arjen K. Lenstra, *Lattice sieving and trial division*, Algorithmic Number Theory (Berlin) (L.M. Adleman and M.D. Huang, eds.), Lecture Notes in Computer Science, vol. 877, Springer-Verlag, 1994, pp. 18–27.

[29] Ian N. Stewart and David O. Tall, *Algebraic number theory*, second ed., Chapman and Hall, London, July 1987.

Vita

Matthew Edward Briggs was born August 29, 1974 in Fairfax, Virginia to Tedford C. Briggs and Beverly D. Briggs. He is an only child who lived in the same house in Springfield, Virginia for 17 years before coming to Virginia Tech in the Fall of 1992. Matthew earned a Bachelor of Science degree in Computer Science from Virginia Tech in the Spring of 1996, also graduating with a double major in Mathematics. Matthew immediately began Graduate study at Virginia Tech in the Summer of 1996 and continued on until the Spring of 1998 when he received a Master of Science degree in Mathematics.

While pursuing his graduate degree at Virginia Tech, Matthew put his Computer Science skills to use working part-time at a local software company in the Virginia Tech Corporate Research Center.