Chapter 7

Introduction to Elliptic Curves

7.1 Basic Definitions

7.1.1 Introduction

The aim of this chapter is to give a brief survey of results, essentially without proofs, about elliptic curves, complex multiplication and their relations to class groups of imaginary quadratic fields. A few algorithms will be given (in Section 7.4, so as not to interrupt the flow of the presentation), but, unlike other chapters, the main emphasis will be on the theory (some of which will be needed in the next chapters). We also describe the superb landscape that is emerging in this theory, although much remains conjectural. It is worth noting that many of the recent advances on the subject (in particular the Birch and Swinnerton-Dyer conjecture) were direct consequences of number-theoretical experiments. This lends further support to the claim that number theory, even in its sophisticated areas, is an experimental as well as a theoretical science.

As elsewhere this book, we have tried to keep the exposition as self-contained as possible. However, for mastering this information, it would be useful if the reader had some knowledge of complex variables and basic algebraic geometry. Nonetheless, the material needed for the applications in the later chapters is fully described here.

As suggestions for further reading, I heartily recommend Silverman's books [Sil] and [Sil3], as well as [Cas], [Hus], [Ire-Ros], [Lang3] and [Shi]. Finally, the algorithms and tables contained in [LN476] (commonly called Antwerp IV) and [Cre] are invaluable.

7.1.2 Elliptic Integrals and Elliptic Functions

Historically, the word elliptic (in the modern sense) came from the theory of elliptic integrals, which occur in many problems, for example in the computation of the length of an arc of an ellipse (whence the name), or in physical problems such as the movement of a pendulum. Such integrals are of the form

$$\int R(x,y)\,dx,$$

where $R(x,y)$ is a rational function in x and y, and y^2 is a polynomial in x of degree 3 or 4 having no multiple root. It is not our purpose here to explain the

theory of these integrals (for this see e.g. [W-W], Ch. XXII). They have served as a motivation for the theory of elliptic *functions*, developed in particular by Abel, Jacobi and Weierstraß.

Elliptic functions can be defined as inverse functions of elliptic integrals, but the main property that interests us here is that these functions $f(x)$ are doubly periodic. More precisely we have:

Definition 7.1.1. *An* elliptic function *is a meromorphic function $f(x)$ on the whole complex plane, which is doubly periodic, i.e. such that there exist complex numbers ω_1 and ω_2 such that $\omega_1/\omega_2 \notin \mathbb{R}$ and for all x which is not a pole, $f(x+\omega_1) = f(x+\omega_2) = f(x)$.*

If

$$L = \{m\omega_1 + n\omega_2 | m, n \in \mathbb{Z}\}$$

is the lattice generated by ω_1 and ω_2, it is clear that f is elliptic if and only if $f(x+\omega) = f(x)$ for all $x \in \mathbb{C}$ and all $\omega \in L$. The lattice L is called the period lattice of f. It is clear that every element of $\mathbb{C}$ is equivalent modulo a translation by an element of L to a unique element of the set $F = \{x\omega_1 + y\omega_2,\ 0 \leq x, y < 1\}$. Such a set will be called a *fundamental domain* for $\mathbb{C}/L$.

Standard residue calculations immediately show the following properties:

Theorem 7.1.2. *Let $f(x)$ be an elliptic function with period lattice L, let $\{z_i\}$ be the set of zeros and poles of f in a fundamental domain for $\mathbb{C}/L$, and n_i be the order of f at z_i ($n_i > 0$ when z_i is a zero, $n_i < 0$ if z_i is a pole). Then*

(1) *The sum of the residues of f in a fundamental domain is equal to 0.*
(2) *$\sum_i n_i = 0$, in other words f has as many zeros as poles (counted with multiplicity).*
(3) *If f is non-constant, counting multiplicity, f must have at least 2 poles (and hence 2 zeros) in a fundamental domain.*
(4) *$\sum_i n_i z_i \in L$. Note that this makes sense since z_i is defined modulo L.*

Note that the existence of non-constant elliptic functions is not a priori obvious from Definition 7.1.1. In fact, we have the following general theorem, due to Abel and Jacobi:

Theorem 7.1.3. *Assume that z_i and n_i satisfy the above properties. Then there exists an elliptic function f with zeros and poles at z_i of order n_i.*

The simplest construction of non-constant elliptic functions is due to Weierstraß. One defines

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega \in L \setminus \{0\}} \left(\frac{1}{(z+\omega)^2} - \frac{1}{\omega^2} \right),$$

and one easily checks that this is an absolutely convergent series which defines an elliptic function with a double pole at 0. Since non-constant elliptic functions must have poles, it is then a simple matter to check that if we define

$$g_2 = 60 \sum_{\omega \in L \setminus \{0\}} \frac{1}{\omega^4} \quad \text{and} \quad g_3 = 140 \sum_{\omega \in L \setminus \{0\}} \frac{1}{\omega^6},$$

then $\wp(z)$ satisfies the following differential equation:

$$\wp'^2 = 4\wp^3 - g_2\wp - g_3 .$$

In more geometric terms, one can say that the map

$$z \mapsto \begin{cases} (\wp(z) : \wp'(z) : 1) & \text{for } z \notin L \\ (0 : 1 : 0) & \text{for } z \in L \end{cases}$$

from $\mathbb{C}$ to the projective complex plane gives an isomorphism between the torus $\mathbb{C}/L$ and the projective algebraic curve $y^2t = 4x^3 - g_2xt^2 - g_3t^3$. This is in fact a special case of a general theorem of Riemann which states that all compact Riemann surfaces are algebraic. Note that it is easy to prove that the field of elliptic functions is generated by $\wp$ and $\wp'$ subject to the above algebraic relation.

Since $\mathbb{C}/L$ is non-singular, the corresponding algebraic curve must also be non-singular, and this is equivalent to saying that the discriminant

$$\Delta = 16(g_2^3 - 27g_3^2)$$

of the cubic polynomial is non-zero. This leads directly to the definition of elliptic curves.

7.1.3 Elliptic Curves over a Field

From the preceding section, we see that there are at least two ways to generalize the above concepts to an arbitrary field: we could define an elliptic curve as a curve of genus 1 or as a non-singular plane cubic curve. Luckily, the Riemann-Roch theorem shows that these two definitions are equivalent, hence we set:

Definition 7.1.4. *Let K be a field. An* elliptic curve over K *is a non-singular projective plane cubic curve E together with a point with coordinates in K. The (non-empty) set of projective points which are on the curve and with coordinates in K will be called the set of K-rational points of E and denoted $E(K)$.*

Up to a suitable birational transformation, it is a simple matter to check that such a curve can always be given by an equation of the following (affine) type:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

the point defined over K being the (unique) point at infinity, and hence this can be taken as an alternative definition of an elliptic curve (see Algorithm 7.4.10 for the explicit formulas for the transformation). This will be called a (generalized) Weierstraß equation for the curve.

Note that this equation is not unique. Over certain number fields K such as $\mathbb{Q}$, it can be shown however that there exists an equation which is minimal, in a well defined sense. We will call it *the* minimal Weierstraß equation of the curve. Note that such a minimal equation does not necessarily exist for any number field K. For example, it can be shown (see [Sil], page 226) that the elliptic curve $y^2 = x^3 + 125$ has no minimal Weierstraß equation over the field $\mathbb{Q}(\sqrt{-10})$.

Theorem 7.1.5. *An elliptic curve over $\mathbb{C}$ has the form $\mathbb{C}/L$ where L is a lattice. In other words, if g_2 and g_3 are any complex numbers such that $g_2^3 - 27g_3^2 \neq 0$, then there exist ω_1 and ω_2 with $\mathrm{Im}(\omega_1/\omega_2) > 0$ and $g_2 = 60\sum_{(m,n)\neq(0,0)}(m\omega_1 + n\omega_2)^{-4}$, $g_3 = 140\sum_{(m,n)\neq(0,0)}(m\omega_1 + n\omega_2)^{-6}$.*

A fundamental property of elliptic curves is that they are commutative algebraic groups. This is true over any base field. Over $\mathbb{C}$ this follows immediately from Theorem 7.1.5. The group law is then simply the quotient group law of $\mathbb{C}$ by L. On the other hand, it is not difficult to prove the addition theorem for the Weierstraß $\wp$ function, given by:

$$\wp(z_1+z_2) = \begin{cases} -\wp(z_1) - \wp(z_2) + \frac{1}{4}\left(\frac{\wp'(z_1)-\wp'(z_2)}{\wp(z_1)-\wp(z_2)}\right)^2, & \text{if } z_1 \neq z_2; \\ -2\wp(z_1) + \frac{1}{4}\left(\frac{\wp''(z_1)}{\wp'(z_1)}\right)^2, & \text{if } z_1 = z_2. \end{cases}$$

From this and the isomorphism given by the map $z \mapsto (\wp(z), \wp'(z))$, one obtains immediately:

Proposition 7.1.6. *Let $y^2 = 4x^3 - g_2x - g_3$ be the equation of an elliptic curve. The neutral element for the group law is the point at infinity $(0:1:0)$. The inverse of a point (x_1, y_1) is the point $(x_1, -y_1)$ i.e. the symmetric point with respect to the x-axis. Finally, if $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two non-opposite points on the curve, their sum $P_3 = (x_3, y_3)$ is given by the following formulas. Set*

$$m = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}, & \text{if } P_1 \neq P_2; \\ \frac{12x_1^2 - g_2}{2y_1}, & \text{if } P_1 = P_2. \end{cases}$$

Then

$$x_3 = -x_1 - x_2 + m^2/4, \quad y_3 = -y_1 - m(x_3 - x_1).$$

It is easy to see that this theorem enables us to define an addition law on an elliptic curve over any base field of characteristic zero, and in fact in any characteristic different from 2 and 3. Furthermore, it can be checked that this indeed defines a group law.

More generally one can define such a law over any field, in the following way.

Proposition 7.1.7. *Let*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

be the equation of an elliptic curve defined over an arbitrary base field. Define the neutral element as the point at infinity $(0 : 1 : 0)$*, the opposite of a point* (x_1, y_1) *as the point* $(x_1, -y_1 - a_1x_1 - a_3)$*. Finally, if* $P_1 = (x_1, y_1)$ *and* $P_2 = (x_2, y_2)$ *are two non-opposite points on the curve, define their sum* $P_3 = (x_3, y_3)$ *by the following. Set*

$$m = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2}, & \text{if } P_1 \neq P_2; \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{if } P_1 = P_2, \end{cases}$$

and put

$$x_3 = -x_1 - x_2 - a_2 + m(m + a_1), \quad y_3 = -y_1 - a_3 - a_1x_3 + m(x_1 - x_3).$$

Then these formulas define an (algebraic) Abelian group law on the curve.

The only non-trivial thing to check in this theorem is the associativity of the law. This can most easily be seen by interpreting the group law in terms of divisors, but we will not do this here.

The geometric interpretation of the formulas above is the following. Let P_1 and P_2 be points on the (projective) curve. The line D from P_1 to P_2 (the tangent to the curve if $P_1 = P_2$) intersects the curve at a third point R, say. Then, if O is the point at infinity on the curve, the sum of P_1 and P_2 is the third point of intersection with the curve of the line from O to R. One checks easily that this leads to the above formulas.

For future reference, given a general equation as above, we define the following quantities:

$$\begin{aligned} &b_2 = a_1^2 + 4a_2, \quad b_4 = a_1a_3 + 2a_4 \\ &b_6 = a_3^2 + 4a_6, \quad b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\ &c_4 = b_2^2 - 24b_4, \quad c_6 = -b_2^3 + 36b_2b_4 - 216b_6 \\ &\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6, \quad j = c_4^3/\Delta \\ &\omega = dx/(2y + a_1x + a_3) = dy/(3x^2 + 2a_2x + a_4 - a_1y). \end{aligned} \tag{7.1}$$

Then it is easy to see that if we set $Y = 2y+a_1x+a_3$, on a field of characteristic different from 2, the equation becomes

$$Y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6\,.$$

Setting $X = x + b_2/12$, if the characteristic of the field is different from 2 and 3 the equation becomes

$$Y^2 = 4X^3 - (c_4/12)X - (c_6/216).$$

7.1.4 Points on Elliptic Curves

Consider an abstract equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where the coefficients a_i are in $\mathbb{Z}$. Since for any field K there exists a natural homomorphism from $\mathbb{Z}$ to K, this equation can be considered as defining a curve over any field K. Note that even if the initial curve was non-singular, in positive characteristic the curve can become singular.

We shall consider successively the case where $K = \mathbb{R}$, $K = \mathbb{F}_q$, where q is a power of a prime p, and $K = \mathbb{Q}$.

Elliptic Curves over $\mathbb{R}$. In the case where the characteristic is different from 2 and 3, the general equation can be reduced to the following Weierstraß form:

$$y^2 = x^3 + a_4x + a_6\,.$$

(We could put a 4 in front of the x^3 as in the equation for the $\wp$ function, but this introduces unnecessary constant factors in the formulas). The discriminant of the cubic *polynomial* is $-(4a_4^3 + 27a_6^2)$, however the y^2 term must be taken into account, and general considerations show that one must take

$$-16(4a_4^3 + 27a_6^2)$$

as the definition of the discriminant of the elliptic curve.

Several cases can occur. Let $Q(x) = x^3+a_4x+a_6$ and $\Delta = -16(4a_4^3+27a_6^2)$.

(1) $\Delta < 0$. Then the equation $Q(x) = 0$ has only one real root, and the graph of the curve has only one connected component.

(2) $\Delta > 0$. Then the equation $Q(x) = 0$ has three distinct real roots, and the graph of the curve has two connected components: a non-compact one, which is the component of the zero element of the curve (i.e. the point at infinity), and a compact one, oval shaped.

From the geometric construction of the group law, one sees that the roots of $Q(x) = 0$ are exactly the points of order 2 on the curve (the points of order 3 correspond to the inflection points).

(3) $\Delta = 0$. The curve is no longer an elliptic curve, since it now has a singular point. This case splits into three sub-cases. Since the polynomial $Q(x)$ has at least a double root, write

$$Q(x) = (x-a)^2(x-b) \ .$$

Note that $2a + b = 0$.

(3a) $a > b$. Then the curve has a unique connected component, which has a double point at $x = a$. The tangents at the double point have distinct real slopes.

(3b) $a < b$. Then the curve has two connected components: a non-compact one, and the single point of coordinates $(a, 0)$. In fact this point is again a double point, but with distinct *purely imaginary* tangents.

(3c) $a = b$. (In this case $a = b = 0$ since $2a + b = 0$). Then the curve has a cusp at $x = 0$, i.e. the tangents at the singular point are the same.

See Fig. 7.1 for the different possible cases. Note that case (1) is subdivided into the case where the curve does not have any horizontal tangent ($a_4 > 0$), and the case where it does ($a_4 \leq 0$).

In case 3, one says that the curve is a degenerate elliptic curve. One easily checks that the group law still exists, but on the curve minus the singular point. This leads to the following terminology: in cases 3a, the group is naturally isomorphic to $\mathbb{R}^*$, and this is called the case of split multiplicative degeneracy. In case 3b, the group is isomorphic to the group S^1 of complex numbers of modulus equal to 1, and this is called non-split multiplicative degeneracy. Finally, in case 3c, the group is isomorphic to the additive group $\mathbb{R}$, and this case is called additive degeneracy.

These notions can be used, not only for $\mathbb{R}$, but for any base field K. In that case, the condition $a > b$ is replaced by $a - b$ is a (non-zero) square in K.

Elliptic Curves over a Finite Field. To study curves (or more general algebraic objects) over $\mathbb{Q}$, it is very useful to study first the reduction of the curve modulo primes. This leads naturally to elliptic curves over $\mathbb{F}_p$, and more generally over an arbitrary finite field $\mathbb{F}_q$, where q is a power of p. Note that when one reduces an elliptic curve mod p, the resulting curve over $\mathbb{F}_p$ may be singular, hence no longer an elliptic curve. Such p are called primes of bad reduction, and are finite in number since they must divide the discriminant of the curve. According to the terminology introduced in the case of $\mathbb{R}$, we will say that the reduction mod p is (split or non-split) multiplicative or additive, according to the type of degeneracy of the curve over $\mathbb{F}_p$. The main theorem concerning elliptic curves over finite fields, due to Hasse, is as follows:

Theorem 7.1.8 (Hasse). *Let p be a prime, and E an elliptic curve over $\mathbb{F}_p$. Then there exists an imaginary quadratic integer α_p such that*

(1) *If $q = p^n$ then*

$$|E(\mathbb{F}_q)| = q + 1 - {\alpha_p}^n - \overline{\alpha_p}^n$$

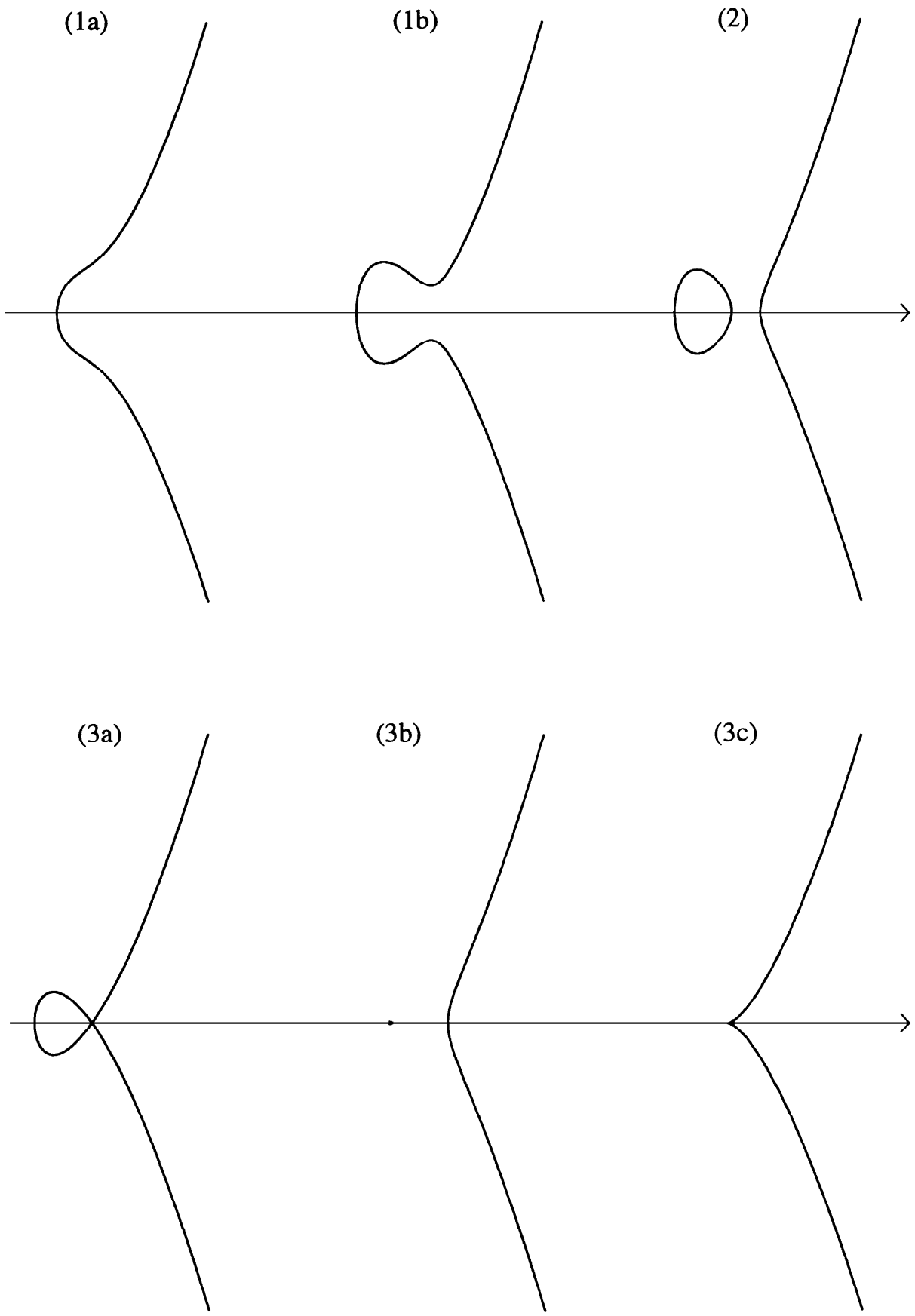

Figure 7.1. Non-Degenerate and Degenerate Elliptic Curves over $\mathbb{R}$.

(2)
$$\alpha_p\overline{\alpha_p} = p, \text{ or equivalently } |\alpha_p| = \sqrt{p}\,.$$

(3) *In particular, we have*
$$|E(\mathbb{F}_p)| = p + 1 - a_p \quad \text{with } |a_p| < 2\sqrt{p},$$
and α_p is a root of the equation
$$\alpha_p{}^2 - a_p\alpha_p + p = 0\,.$$

The numbers a_p are very important and are (conjecturally) coefficients of a modular form of weight 2. We will come back to this subject in Section 7.3.

The second important result gives some information on the group structure of $E(\mathbb{F}_q)$, and is as follows.

Proposition 7.1.9. *If E is an elliptic curve over a finite field $\mathbb{F}_q$, then $E(\mathbb{F}_q)$ is either cyclic or isomorphic to a product of two cyclic groups. Furthermore, in the case where it is not cyclic, if we write $E(\mathbb{F}_q) \simeq \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z}$ with $d_1 \mid d_2$, then $d_1 \mid q-1$.*

Elliptic Curves over $\mathbb{Q}$. From a number theorist's point of view, this is of course the most interesting base field. The situation in this case and in the case of more general number fields is much more difficult. The first basic theorem, due to Mordell and later generalized by Weil to the case of number fields and of Abelian varieties, is as follows:

Theorem 7.1.10 (Mordell). *Let E be an elliptic curve over $\mathbb{Q}$. The group of points of E with coordinates in $\mathbb{Q}$ (denoted naturally $E(\mathbb{Q})$) is a finitely generated Abelian group. In other words,*
$$E(\mathbb{Q}) \simeq E(\mathbb{Q})_{\text{tors}} \oplus \mathbb{Z}^r,$$
where r is a non-negative integer called the rank of the curve, and $E(\mathbb{Q})_{\text{tors}}$ is the torsion subgroup of $E(\mathbb{Q})$, which is a finite Abelian group.

The torsion subgroup of a given elliptic curve is easy to compute. On the other hand the study of possible torsion subgroups for elliptic curves over $\mathbb{Q}$ is a difficult problem, solved only in 1977 by Mazur ([Maz]). His theorem is as follows:

Theorem 7.1.11 (Mazur). *Let E be an elliptic curve over $\mathbb{Q}$. The torsion subgroup $E(\mathbb{Q})_{\text{tors}}$ of E can be isomorphic only to one of the 15 following groups:*
$$\mathbb{Z}/m\mathbb{Z} \quad \text{for } 1 \le m \le 10 \text{ or } m = 12,$$

$$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2m\mathbb{Z} \quad \text{for } 1 \leq m \leq 4\,.$$

In particular, its cardinality is at most 16.

Note that all of the 15 groups above do occur for an infinite number of non-isomorphic elliptic curves. The corresponding theorem for all quadratic fields (even allowing the discriminant to vary) was proved in 1990 by Kamienny ([Kam]) (with more groups of course), and finally for all number fields in 1994 by Merel ([Mer]).

The other quantity which occurs in Mordell's theorem is the rank r, and is a much more difficult number to compute, even for an individual curve. There is no known mathematically proven algorithm to compute r in general. Even the apparently simpler question of deciding whether r is zero or not (or equivalently whether the curve has a finite or an infinite number of rational points) is still not solved. This is the subject of active research, and we will come back in more detail to this question in Section 7.4.

Let us give an example of a down to earth application of Mordell's theorem. Consider the curve

$$y^2 = x^3 - 36x\,.$$

It is easy to show (see Exercise 3) that the only torsion points are the points of order 1 or 2, i.e. the point at infinity and the three points $(0,0), (6,0), (-6,0)$. But the point $(-2,8)$ is also on the curve. Therefore we must have $r > 0$, hence an infinite number of points, a fact which is not a priori evident. What Mordell's theorem tells us is that r is *finite*, and in fact one can show in this case that $r = 1$, and that the *only* rational points on the curve are integral multiples of the point $(-2,8)$ added to one of the four torsion points.

This curve is in fact closely related to the so-called congruent number problem, and the statement that we have just made means, in this context, that there exists an infinite number of non-equivalent right angled triangles with all three sides rational and area equal to 6, the simplest one (corresponding to the point $(-2,8)$) being the well known $(3,4,5)$ Pythagorean triangle.

As an exercise, the reader can check that twice the point $(-2,8)$ is the point $\left(\frac{25}{4}, \frac{35}{8}\right)$, and that this corresponds to the right-angled triangle of area 6 with sides $\left(\frac{120}{7}, \frac{7}{10}, \frac{1201}{70}\right)$. See [Kob] for the (almost) complete story on the congruent number problem.

7.2 Complex Multiplication and Class Numbers

In this section, we will study maps between elliptic curves. We begin by the case of curves over $\mathbb{C}$.

7.2.1 Maps Between Complex Elliptic Curves

Recall that a complex elliptic curve E has the form $\mathbb{C}/L$ where L is a lattice. Let $E = \mathbb{C}/L$ and $E' = \mathbb{C}/L'$ be two elliptic curves. A map ϕ from E to E' is by definition a holomorphic $\mathbb{Z}$-linear map from E to E'. Since $\mathbb{C}$ is the universal covering of E', ϕ lifts to a holomorphic $\mathbb{Z}$-linear map f from $\mathbb{C}$ to $\mathbb{C}$, and such a map has the form $f(z) = \alpha z$ for some complex number α, which induces a map from E to E' iff $\alpha L \subset L'$. Thus we have:

Proposition 7.2.1. *Let $E = \mathbb{C}/L$ and $E' = \mathbb{C}/L'$ be two elliptic curves over $\mathbb{C}$. Then*

(1) *E is isomorphic to E' if and only if $L' = \alpha L$ for a certain non-zero complex number α.*
(2) *The set of maps from E to E' can be identified with the set of complex numbers α such that $\alpha L \subset L'$. In particular, the set $\mathrm{End}(E)$ of endomorphisms of E is a commutative ring isomorphic to the set of α such that $\alpha L \subset L$.*

In terms of the Weierstraß equation of the curves, this theorem gives the following. Recall that the equation of E (resp E') is $y^2 = 4x^3 - g_2 x - g_3$ (resp. $y^2 = 4x^3 - g_2' x - g_3'$) where

$$g_2 = 60 \sum_{\omega \in L \setminus \{0\}} \omega^{-4}, \quad g_3 = 140 \sum_{\omega \in L \setminus \{0\}} \omega^{-6},$$

and similarly for g_2' and g_3'. Hence, the first part of the theorem says that if $E \simeq E'$, there exists α such that

$$g_2' = \alpha^{-4} g_2 \ , \quad g_3' = \alpha^{-6} g_3 \, .$$

The converse is also clear from the Weierstraß equation. Now, since E is a non-singular curve, the discriminant $g_2^3 - 27 g_3^2$ is non-zero, so we can define

$$j(E) = 1728 g_2^3 / (g_2^3 - 27 g_3^2),$$

and we obtain:

Proposition 7.2.2. *The function $j(E)$ characterizes the isomorphism class of E over $\mathbb{C}$. More precisely, $E \simeq E'$ if and only if $j(E) = j(E')$.*

The quantity $j(E)$ is called the modular invariant of the elliptic curve E. The number $1728 = 12^3$ will be explained later. Although we have been working over $\mathbb{C}$, Proposition 7.2.2 is still valid over any algebraically closed field of characteristic different from 2 and 3 (it is also valid in characteristic 2 or 3, for a slightly generalized definition of $j(E)$). On the other hand, it is

false if the field is not algebraically closed (consider for example $y^2 = 4x^3 - 4x$ and $y^2 = 4x^3 + 4x$ over $\mathbb{R}$).

Remark. It is easy to construct an elliptic curve with a given modular invariant j. We give the formulas when the characteristic is different from 2 and 3 since we have not given the definition otherwise.

(1) If $j = 0$, one can take $y^2 = x^3 - 1$.
(2) If $j = 1728$, one can take $y^2 = x^3 - x$.
(3) Otherwise, one sets $c = j/(j - 1728)$, and then one can take $y^2 = x^3 - 3cx + 2c$. (If one wants equations with a coefficient of 4 in front of x^3, multiply by 4 and replace y by $y/2$.)

Now let $E = \mathbb{C}/L$ be an elliptic curve over $\mathbb{C}$. Then, as a $\mathbb{Z}$-module, L can be generated by two $\mathbb{R}$-linearly independent complex numbers ω_1 and ω_2, and by suitably ordering them, we may assume that $\operatorname{Im}\tau > 0$, where $\tau = \omega_1/\omega_2$. Since multiplying a lattice by a non-zero complex number does not change the isomorphism class of E, we have $j(E) = j(E_\tau)$, where $E_\tau = \mathbb{C}/L_\tau$ and L_τ is the lattice generated by 1 and τ. By abuse of notation, we will write $j(\tau) = j(E_\tau)$. This defines a complex function j on the upper half-plane $\mathcal{H} = \{\tau \in \mathbb{C},\ \operatorname{Im}\tau > 0\}$. If a, b, c and d are integers such that $ad - bc = 1$ (i.e. if $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$), then the lattice generated by $a\tau + b$ and $c\tau + d$ is equal to L_τ. This implies the *modular invariance* of $j(\tau)$:

Theorem 7.2.3. *For any* $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$, *we have*

$$j\left(\frac{a\tau + b}{c\tau + d}\right) = j(\tau).$$

In particular, $j(\tau)$ is periodic of period 1. Hence it has a Fourier expansion, and one can prove the following theorem:

Theorem 7.2.4. *There exist positive integers* c_n *such that, if we set* $q = e^{2i\pi\tau}$, *we have for all complex* τ *with* $\operatorname{Im}\tau > 0$:

$$j(\tau) = \frac{1}{q} + 744 + \sum_{n \geq 1} c_n q^n.$$

The factor 1728 used in the definition of j is there to avoid denominators in the Fourier expansion of $j(\tau)$, and more precisely to have a residue equal to 1 at infinity (the local variable at infinity being taken to be q). These theorems show that j is a meromorphic function on the compactification (obtained by adding a point at infinity) of the quotient $\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})$.

Proposition 7.2.5. *The function j is a one-to-one mapping from the compactification of $\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})$ onto the projective complex plane $\mathbb{P}_1(\mathbb{C})$ (which is naturally isomorphic to the Riemann sphere S^2). In other words, $j(\tau)$ takes once and only once every possible value (including infinity) on $\overline{\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})}$.*

Note that this proposition is obtained essentially by combining the remark made after Proposition 7.2.2 (surjectivity) with Proposition 7.2.1 (injectivity).

Since the field of meromorphic functions on the sphere is the field of rational functions, we deduce that the field of *modular functions*, i.e. meromorphic functions which are meromorphic at infinity and invariant under $\mathrm{SL}_2(\mathbb{Z})$, is the field of rational functions in j. In particular, modular functions which are holomorphic outside the point at infinity of the Riemann sphere are simply polynomials in j. Finally, if we want to have such a function which is one to one as in Theorem 7.2.5, the only possibilities are linear polynomials $aj + b$. As mentioned above, the constant 1728 has been chosen so that the residue at infinity is equal to one. If we want to keep this property, we must have $a = 1$. This leaves only the possibility $j + b$ for a function having essentially the same properties as j. In other words, the only freedom that we really have in the choice of the modular function j is the constant term 744 in its Fourier expansion.

Although it is a minor point, I would like to say that the normalization of j with constant term 744 is not the correct one for several reasons. The "correct" constant should be 24, so the "correct" j function should in fact be $j - 720$. Maybe the most natural reason is as follows: there exists a rapidly convergent series due to Rademacher for the Fourier coefficients c_n of j. For $n = 0$, this series gives 24, not 744. Other good reasons are due to Atkin and Zagier (unpublished).

7.2.2 Isogenies

We now come back to the case of elliptic curves over an arbitrary field.

Definition 7.2.6. *Let E and E' be two elliptic curves defined over a field K. An* isogeny *from E to E' is a map of algebraic curves from E to E' sending the zero element of E to the zero element of E'. The curves are said to be isogenous if there exists a non-constant isogeny from E to E'.*

The following theorem summarizes the main properties of non-constant isogenies:

Theorem 7.2.7. *Let ϕ be a non-constant isogeny from E to E'. Then:*

(1) *If K is an algebraically closed field, ϕ is a surjective map.*
(2) *ϕ is a finite map, in other words the fiber over any point of E' is constant and finite.*

(3) *ϕ preserves the group laws of the elliptic curves (note that this was not required in the definition), i.e. it is a map of algebraic groups.*

From these properties, one can see that ϕ induces an injective map from the corresponding function field of E' to that of E (over some algebraic closure of the base field). The degree of the corresponding field extensions is finite and called the degree of ϕ.

Note that if the above extension of fields is separable, for example if the base field has characteristic zero, then the degree of ϕ is also equal to the cardinality of a fiber, i.e. to the cardinality of its kernel $\phi^{-1}(O)$, but this is not true in general.

Theorem 7.2.8. *Let E be an elliptic curve over a field K, and let m be a positive integer. Then the map $[m]$ (multiplication by m) is an endomorphism of E with the following properties:*

(1) $\deg[m] = m^2$.
(2) *Let $E[m]$ denote the kernel of $[m]$ in some algebraic closure of K, i.e. the group of points of order dividing m. If the characteristic of K is prime to m (or if it is equal to 0), we have*

$$E[m] \simeq (\mathbb{Z}/m\mathbb{Z}) \times (\mathbb{Z}/m\mathbb{Z}).$$

Another important point concerning isogenies is the following:

Theorem 7.2.9. *Let ϕ be an isogeny from E to E'. There exists a unique isogeny $\hat{\phi}$ from E' to E called the dual isogeny, such that*

$$\hat{\phi} \circ \phi = [m],$$

where m is the degree of ϕ. In addition, we also have

$$\phi \circ \hat{\phi} = [m]',$$

where $[m]'$ denotes multiplication by m on E'.

Note also the following:

Theorem 7.2.10. *Let E be an elliptic curve and Φ a finite subgroup of E. Then there exists an elliptic curve E' and an isogeny ϕ from E to E' whose kernel is equal to Φ. The elliptic curve E' is well defined up to isomorphism and is denoted E/Φ.*

We end this section by giving a slightly less trivial example of an isogeny: Let E and E' be two elliptic curves over a field of characteristic different from 2, given by the equations

$$y^2 = x^3 + ax^2 + bx \quad \text{and } y^2 = x^3 - 2ax^2 + (a^2 - 4b)x\,,$$

where we assume that b and $a^2 - 4b$ are both non-zero. Then the map ϕ from E to E' given by

$$\phi(x,y) = \left(\frac{y^2}{x^2}, \frac{y(x^2-b)}{x^2}\right)$$

is an isogeny of degree 2 with kernel $\{O, (0,0)\}$.

7.2.3 Complex Multiplication

Let E be an elliptic curve. To make life simpler, we will assume that the base field has characteristic zero. We have seen that the maps $[m]$ are elements of $\mathrm{End}(E)$. Usually, they are the only ones, and since they are distinct, $\mathrm{End}(E) \simeq \mathbb{Z}$. It may however happen that $\mathrm{End}(E)$ is larger than $\mathbb{Z}$.

Definition 7.2.11. *We say that E has* complex multiplication *if* $\mathrm{End}(E)$ *contains elements other than $[m]$, i.e. if as a ring it is strictly larger than $\mathbb{Z}$.*

The theory of complex multiplication is vast, and we can just give a glimpse of its contents. The first result is as follows:

Proposition 7.2.12. *Let E be an elliptic curve defined over a field of characteristic zero, and assume that E has complex multiplication. Then the ring* $\mathrm{End}(E)$ *is an order in an imaginary quadratic field, i.e. has the form $\mathbb{Z} + \mathbb{Z}\tau$ where τ is a complex number with positive imaginary part and which is an algebraic integer of degree 2 (that is, satisfies an equation of the form*

$$\tau^2 - s\tau + n = 0\,,$$

with s and n in $\mathbb{Z}$ and $s^2 - 4n < 0$).

Proof. We shall give the proof in the case where the base field is $\mathbb{C}$. Then $E \simeq \mathbb{C}/L$ for a certain lattice L, and we know that $\mathrm{End}(E)$ is canonically isomorphic to the set of α such that $\alpha L \subset L$. After division by one of the generators of L, we can assume that L is generated by 1 and τ for a certain $\tau \in \mathcal{H}$, where we recall that $\mathcal{H}$ is the upper half-plane. Then if α stabilizes L, there must exist integers a, b, c and d such that $\alpha = a + b\tau$, $\alpha\tau = c + d\tau$. In other words, α is an eigenvalue of the matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, hence is an algebraic integer of degree 2 (with $s = a + d$, $n = ad - bc$). Since $\alpha = a + b\tau$, this shows that $\mathbb{Q}(\tau) = \mathbb{Q}(\alpha)$ is a fixed imaginary quadratic extension k of $\mathbb{Q}$, and hence $\mathrm{End}(E)$ is (canonically isomorphic to) a subring of $\mathbb{Z}_k$, the ring of integers of k, and hence is an order in k if it is larger than $\mathbb{Z}$. □

Example. The curves $y^2 = x^3 - ax$ all have complex multiplication by $\mathbb{Z}[i]$ (map (x, y) to $(-x, iy)$). Similarly, the curves $y^2 = x^3 + b$ all have complex multiplication by $\mathbb{Z}[\rho]$, where ρ is a primitive cube root of unity (map (x, y) to $(\rho x, y)$). For a less trivial example, one can check that the curve

$$y^2 = x^3 - (3/4)x^2 - 2x - 1$$

has complex multiplication by $\mathbb{Z}[\omega]$, where $\omega = \frac{1+\sqrt{-7}}{2}$, multiplication by ω sending (x, y) to (u, v), where

$$u = \omega^{-2}\frac{x^2 - \omega}{x - a}$$

$$v = \omega^{-3}y\frac{x^2 - 2ax + \omega}{(x - a)^2},$$

where we have set $a = (\omega - 3)/4$ (I thank D. Bernardi for these calculations). For a simple algorithm which makes these computations easy to perform see [Star].

Remark. Note that if the base field is a finite field, $\mathrm{End}(E)$ is either isomorphic to an order in an imaginary quadratic field or to the maximal order in a definite quaternion algebra of dimension 4 over $\mathbb{Z}$. In this last case, which is the only case where $\mathrm{End}(E)$ is non-commutative, we say that the elliptic curve E is *supersingular.*

The next theorem concerning complex multiplication is as follows:

Theorem 7.2.13. *Let τ be a quadratic algebraic number with positive imaginary part. Then the elliptic curve $E_\tau = \mathbb{C}/(\mathbb{Z}+\mathbb{Z}\tau)$ has complex multiplication by an order in the quadratic field $\mathbb{Q}(\tau)$, and the j-invariant $j(E_\tau) = j(\tau)$ is an algebraic integer.*

Note that although the context (and the proof) of this theorem involves elliptic curves, its statement is simply that a certain explicit function $j(\tau)$ on $\mathcal{H}$ takes algebraic integer values at quadratic imaginary points.

Examples. Here are a few selected values of j.

$$
\begin{aligned}
j((1+i\sqrt{3})/2) &= 0 = 1728 - 3(24)^2 \\
j(i) &= 1728 = 12^3 = 1728 - 4(0)^2 \\
j((1+i\sqrt{7})/2) &= -3375 = (-15)^3 = 1728 - 7(27)^2 \\
j(i\sqrt{2}) &= 8000 = 20^3 = 1728 + 8(28)^2 \\
j((1+i\sqrt{11})/2) &= -32768 = (-32)^3 = 1728 - 11(56)^2 \\
j((1+i\sqrt{19})/2) &= -884736 = (-96)^3 = 1728 - 19(216)^2 \\
j((1+i\sqrt{43})/2) &= -884736000 = (-960)^3 = 1728 - 43(4536)^2 \\
j((1+i\sqrt{67})/2) &= -147197952000 = (-5280)^3 = 1728 - 67(46872)^2 \\
j((1+i\sqrt{163})/2) &= -262537412640768000 = (-640320)^3 \\
&= 1728 - 163(40133016)^2 \\
j(i\sqrt{3}) &= 54000 = 2(30)^3 = 1728 + 12(66)^2 \\
j(2i) &= 287496 = (66)^3 = 1728 + 8(189)^2 \\
j((1+3i\sqrt{3})/2) &= -12288000 = -3(160)^3 = 1728 - 3(2024)^2 \\
j(i\sqrt{7}) &= 16581375 = (255)^3 = 1728 + 7(1539)^2 \\
j((1+i\sqrt{15})/2) &= \frac{-191025 - 85995\sqrt{5}}{2} \\
&= \frac{1-\sqrt{5}}{2}\left(\frac{75+27\sqrt{5}}{2}\right)^3 = 1728 - 3\left(\frac{273+105\sqrt{5}}{2}\right)^2 \\
j((1+i\sqrt{23})/2) &= -(8207500\theta^2 + 10841250\theta + 6167500) \\
&= -(25\theta^2 + 55\theta + 35)^3 \\
&= 1728 - (3\theta^2 - 4)(406\theta^2 + 511\theta + 273)^2,
\end{aligned}
$$

where θ is the real root of the cubic equation $X^3 - X - 1 = 0$.

The reason for the special values chosen will become clear later.

An amusing consequence of the above results is the following. We know that if $q = e^{2i\pi\tau}$ then $j(\tau) = 1/q + 744 + O(|q|)$. Hence when $|q|$ is very small (i.e. when the imaginary part of τ is large), it can be expected that $j(\tau)$ is well approximated by $1/q + 744$. Taking the most striking example, this implies that $e^{\pi\sqrt{163}}$ should be close to an integer, and that $(e^{\pi\sqrt{163}} - 744)^{1/3}$ should be even closer. This is indeed what one finds:

$$e^{\pi\sqrt{163}} = 262537412640768743.99999999999925007259\ldots$$

$$(e^{\pi\sqrt{163}} - 744)^{1/3} = 640319.99999999999999999999999939031735\ldots$$

Note that by well known transcendence results, although these quantities are very close to integers, they cannot be integers and they are in fact transcendental numbers.

7.2.4 Complex Multiplication and Hilbert Class Fields

The following theorem gives more precise information on the nature of the algebraic integer $j(\tau)$ and will be one of our basic tools in our study of Atkin's primality test (see Section 9.2). We define the *discriminant* of a quadratic number τ as the discriminant of the unique primitive positive definite quadratic form (a, b, c) such that τ is a root of the equation $ax^2+bx+c = 0$.

Theorem 7.2.14. *Let $\tau \in \mathcal{H}$ be a quadratic imaginary number, and let D be its discriminant as just defined. Then $j(\tau)$ is an algebraic integer of degree exactly equal to $h(D)$, where $h(D)$ is the class number of the imaginary quadratic order of discriminant D. More precisely, the minimal polynomial of $j(\tau)$ over $\mathbb{Z}$ is the equation $\prod(X - j(\alpha)) = 0$, where α runs over the quadratic numbers associated to the reduced forms of discriminant D.*

Note that $j(\tau)$ is indeed a root of this polynomial, since any quadratic form of discriminant D is equivalent to a reduced form, and since the j function is $\mathrm{SL}_2(\mathbb{Z})$-invariant. The difficult part of this theorem is that the polynomial has integral coefficients.

I can now explain the reason for the selection of j-values given in the preceding section. From Theorem 7.2.14, we see that $j(\tau)$ is rational (in fact integral) if and only if $h(D) = 1$ (we assume of course that τ is a quadratic number). Hence, by the Heegner-Stark-Baker theorem (see Section 5.3.1), this corresponds to only 9 quadratic fields. There are 4 more corresponding to non-maximal orders: -12 and -27 (in the field $\mathbb{Q}(\sqrt{-3})$), -16 (in the field $\mathbb{Q}(\sqrt{-4})$), and -28 (in the field $\mathbb{Q}(\sqrt{-7})$).

The first 13 values of our little table above correspond to these 13 quadratic orders, and the last two are for $D = -15$ and $D = -23$, which are the first values for which the class number is 2 and 3 respectively.

Now if τ corresponds to a maximal order in an imaginary quadratic field K, Theorem 7.2.14 tells us that the field $H = K(j(\tau))$ obtained by adjoining $j(\tau)$ to K is an algebraic extension of degree $h(D)$ (this is not strictly true: it tells us this for $K = \mathbb{Q}$, but the statement holds nonetheless). Now in fact much more is true: it is a Galois extension, with Abelian Galois group isomorphic to the class group of the imaginary quadratic field K. Furthermore, it is unramified, and it is the maximal Abelian unramified extension of K. By definition, such a field H is called the Hilbert class field of K. One sees that in the case of imaginary quadratic fields, the Hilbert class field can be obtained by adjoining a value of the j-function. This kind of construction is lacking for other types of fields (except of course for $\mathbb{Q}$). See [Shi] for the relevant definitions and theorems about class fields.

A cursory glance at the table of j-values which we have given reveals many other interesting aspects. For example, in most cases, it seems that $j(\tau)$ is a cube. Furthermore, it can be checked that no large prime factors occur in the values of $j(\tau)$ (or of its norm when it is not in $\mathbb{Q}$). These properties

are indeed quite general, with some restrictions. For example, if D is not divisible by 3, then up to multiplication by a unit, $j(\tau)$ is a cube in H. One can also check that (still up to units) $j(\tau) - 1728$ is a square in K if $D \equiv 1 \pmod 4$. Finally, not only the values of $j(\tau)$, but more generally the differences $j(\tau_1) - j(\tau_2)$ have only small prime factors (the case of $j(\tau_1)$ is recovered by taking $\tau_2 = \rho = (-1 + \sqrt{-3})/2$). All these properties have been proved by Gross-Zagier [Gro-Zag1].

The other property of an elliptic curve with complex multiplication, which will also be basic to Atkin's primality test, is that it is easy to compute the number of its points in a finite field, i.e. its L-function (see Section 7.3 for the definition). We state only the special cases which we will need (see [Deu]).

Theorem 7.2.15. *Let E be an elliptic curve with complex multiplication by an imaginary quadratic order of discriminant D, and let p be a prime number. Then we have*

$$|E(\mathbb{F}_p)| = p + 1 - a_p,$$

where a_p is given as follows.

(1) *If p is inert (i.e. if $\left(\frac{D}{p}\right) = -1$), then $a_p = 0$.*
(2) *If p splits into a product of prime* elements, *say $p = \pi\bar{\pi}$, then $a_p = \pi + \bar{\pi}$ for a suitable choice of π.*

Remarks.

(1) If $D < -4$, there exist only two (opposite) choices for π since the order has only 2 units. These choices give two opposite values of a_p, one of these values giving the correct a_p for E, the other one giving the a_p for the curve E "twisted" by a quadratic non-residue (see Section 7.4.3). On the other hand if $D = -4$ or $D = -3$, there exist 4 (resp. 6) choices for π, also corresponding to twisted curves.
(2) If p is ramified or splits into a product of prime ideals which are not principal, then one can still give the value of a_p, but the recipe is more involved. In terms of L-functions, the general result says that there exists a Hecke character ψ on the field $\mathbb{Q}(\sqrt{D})$ such that

$$L(E, s) = L(\psi, s)L(\bar{\psi}, s).$$

7.2.5 Modular Equations

Another remarkable property of the j-function, which is not directly linked to complex multiplication, but rather to the role that j plays as a modular invariant, is that the functions $j(N\tau)$ for N integral (or more generally rational) are algebraic functions of $j(\tau)$. The minimal equation of the form

$\Phi_N(j(\tau), j(N\tau)) = 0$ satisfied by $j(N\tau)$ is called the modular equation of level N. This result is not difficult to prove. We will prove it explicitly in the special case $N = 2$. Set

$$P(X) = (X - j(2\tau))(X - j(\frac{\tau}{2}))(X - j(\frac{\tau+1}{2})) = X^3 - s(\tau)X^2 + t(\tau)X - n(\tau).$$

I claim that the functions s, t and n are polynomials in j. Since they are clearly meromorphic, and in fact holomorphic outside infinity, from Section 7.2.1 it is enough to prove that they are modular functions (i.e. invariant under $\mathrm{SL}_2(\mathbb{Z})$). Since the action of $\mathrm{SL}_2(\mathbb{Z})$ on $\mathcal{H}$ is generated by $\tau \mapsto \tau + 1$, and $\tau \mapsto -1/\tau$, it suffices to show the invariance of s, t and n under these transformations, and this is easily done using the modular invariance of j itself. This shows the existence of a cubic equation satisfied by $j(2\tau)$ over the field $\mathbb{C}(j(\tau))$. If one wants the equation explicitly, one must compute the first few coefficients of the Fourier expansion of $s(\tau)$, $t(\tau)$, and $n(\tau)$, using the Fourier expansion of $j(\tau)$:

$$j(\tau) = \frac{1}{q} + 744 + 196884q + 21493760q^2 + 864299970q^3 + \cdots$$

The result is as follows:

$$s = j^2 - 2^4 3 \cdot 31 j - 2^4 3^4 5^3,$$
$$t = 2^4 3 \cdot 31 j^2 + 3^4 5^3 4027 j + 2^8 3^7 5^6,$$
$$n = -j^3 + 2^4 3^4 5^3 j^2 - 2^8 3^7 5^6 j + 2^{12} 3^9 5^9.$$

This gives as modular polynomial of level 2 the polynomial

$$\Phi_2(X,Y) = X^3 + Y^3 - X^2Y^2 + 2^4 3 \cdot 31(X^2Y + XY^2) - 2^4 3^4 5^3(X^2 + Y^2) + 3^4 5^3 4027 XY + 2^8 3^7 5^6(X + Y) - 2^{12} 3^9 5^9.$$

As we can see from this example, the modular polynomials are symmetric in X and Y. They have many other remarkable properties that tie them closely to complex multiplication and class numbers, but we will not pursue this subject any further here. See for example [Her], [Mah] and [Coh3] for results and more references on the polynomials Φ_N.

7.3 Rank and L-functions

We have seen in Theorem 7.1.10 that if E is an elliptic curve defined over $\mathbb{Q}$, then

$$E(\mathbb{Q}) \simeq E(\mathbb{Q})_{\mathrm{tors}} \oplus \mathbb{Z}^r,$$

where $E(\mathbb{Q})_{\mathrm{tors}}$ is a finite group which is easy to compute for a given curve, and r is an integer called the rank. As has already been mentioned, r is very difficult to compute, even for a specific curve. Most questions here have conjectural answers, but very few are proved. In this section, we try to give some indications on the status of the subject at the time of this writing.

7.3.1 The Zeta Function of a Variety

I heartily recommend reading [Ire-Ros] for detailed and concrete examples on this subject.

After clearing the denominators of the coefficients, we may assume that our curve has coefficients in $\mathbb{Z}$. Now it is a classical technique to look at the equation modulo primes p, and to gather this information to obtain results on the equation over $\mathbb{Q}$ or over $\mathbb{Z}$. This can be done more generally for any smooth projective algebraic variety (and more general objects if needed), and not only for elliptic curves. Although it carries us a little away, I believe it worthwhile to do it in this more general context first.

Let V be a (smooth projective) variety of dimension d, defined by equations with coefficients in $\mathbb{Z}$. For any prime p, we can consider the variety V_p obtained by reducing the coefficients modulo p (it may, of course, not be smooth any more). For any $n \geq 1$, let $N_n(p)$ be the number of points of V_p defined over the finite field $\mathbb{F}_{p^n}$ and consider the following formal power series in the variable T:

$$Z_p(T) = \exp\left(\sum_{n\geq 1} \frac{N_n(p)}{n} T^n\right).$$

Then we have the following very deep theorem, first conjectured by Weil (and proved by him for curves and Abelian varieties, see [Weil]), and proved completely by Deligne in 1974 [Del]:

Theorem 7.3.1. *Let V_p be a smooth projective variety of dimension d over $\mathbb{F}_p$. Then:*

(1) *The series $Z_p(T)$ is a rational function of T, i.e. $Z_p(T) \in \mathbb{Q}(T)$.*
(2) *There exists an integer e (called the Euler characteristic of V_p), such that*

$$Z_p(1/(p^d T)) = \pm p^{de/2} T^e Z_p(T).$$

(3) *The rational function $Z_p(T)$ factors as follows:*

$$Z_p(T) = \frac{P_1(T)\cdots P_{2d-1}(T)}{P_0(T)P_2(T)\cdots P_{2d}(T)},$$

where for all i, $P_i(T) \in \mathbb{Z}[T]$, $P_0(T) = 1 - T$, $P_{2d}(T) = 1 - p^d T$, and for all other i,

$$P_i(T) = \prod_j (1 - \alpha_{ij} T) \quad \text{with } |\alpha_{ij}| = p^{i/2}.$$

The first assertion was actually proved by Dwork a few years before Deligne using relatively elementary methods, but by far the hardest part of this theorem is the last assertion, that $|\alpha_{ij}| = p^{i/2}$. This is called the Riemann hypothesis for varieties over finite fields.

Now given all the local $Z_p(T)$, we can form a global zeta function by setting for s complex with $\mathrm{Re}\, s$ sufficiently large:

$$\zeta(V, s) = \prod_p Z_p(p^{-s}).$$

This should be taken with a grain of salt, since there are some p (finite in number) such that V_p is not smooth. In fact, given the underlying cohomological interpretation of the P_i, it is more reasonable to consider the global L-functions defined by

$$L_i(V, s) = \prod_p P_i(p^{-s})^{-1} \quad \text{for } 0 \leq i \leq 2d,$$

and recover the zeta function as

$$\zeta(V, s) = \prod_{0 \leq i \leq 2d} L_i(V, s)^{(-1)^i}.$$

Very little is known about these general zeta function and L-functions. It is believed (can one say conjectured when so few cases have been closely examined?) that these functions can be analytically continued to meromorphic functions on the whole complex plane. When the local factors at the bad primes p are correctly chosen, they should have a functional equation and the L-functions should satisfy the Riemann hypothesis, i.e. apart from "trivial" zeros, all the other complex zeros of $L_i(V, s)$ should lie on the vertical line $\mathrm{Re}\, s = (i+1)/2$.

One recovers the ordinary Riemann zeta function by taking for V the single point 0. More generally, one can recover the Dedekind zeta function of a number field by taking for V the 0-dimensional variety defined in the projective line by $P(X) = 0$, where P is a monic polynomial with integer coefficients defining the field over $\mathbb{Q}$.

7.3.2 L-functions of Elliptic Curves

Let us now consider the special case where V is an elliptic curve E. In that case, Hasse's Theorem 7.1.8 gives us all the information we need about the number of points of E over a finite field. This leads to the following corollary:

Corollary 7.3.2. *Let E be an elliptic curve over $\mathbb{Q}$, and let p be a prime of good reduction (i.e. such that E_p is still smooth). Then*

$$Z_p(E) = \frac{1 - a_p T + pT^2}{(1-T)(1-pT)},$$

where a_p is as in Theorem 7.1.8.

In fact, Hasse's theorem is simply the special case of the Weil conjectures for elliptic curves (and can be proved quite simply, see e.g. [Sil] pp 134–136).

Ignoring for the moment the question of bad primes, the general definition of zeta and L-functions gives us

$$\zeta(E,s) = \frac{\zeta(s)\zeta(s-1)}{L(E,s)},$$

where

$$L(E,s) = L_1(E,s) = \prod_p (1 - a_p p^{-s} + p^{1-2s})^{-1}.$$

The function $L(E,s)$ will be called the Hasse-Weil L-function of the elliptic curve E. To give a precise definition, we also need to define the local factors at the bad primes p. This can be done, and finally leads to the following definition.

Definition 7.3.3. *Let E be an elliptic curve over $\mathbb{Q}$, and let $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$ be a minimal Weierstraß equation for E (see 7.1.3). When E has good reduction at p, define $a_p = p+1-N_p$ where N_p is the number of (projective) points of E over $\mathbb{F}_p$. If E has bad reduction, define*

$$\epsilon(p) = \begin{cases} 1 & \text{if } E \text{ has split multiplicative reduction at } p; \\ -1 & \text{if } E \text{ has non-split multiplicative reduction at } p; \\ 0 & \text{if } E \text{ has additive reduction at } p. \end{cases}$$

*Then we define the L-*function *of E as follows, for* $\operatorname{Re} s > 3/2$*:*

$$L(E,s) = \prod_{\text{bad } p} \frac{1}{1-\epsilon(p)p^{-s}} \prod_{\text{good } p} \frac{1}{1 - a_p p^{-s} + p^{1-2s}}.$$

Note that in this definition it is crucial to take a minimal Weierstraß equation for E: taking another equation could increase the number of primes of bad reduction, and hence change a finite number of local factors. On the other hand, one can prove that $L(E,s)$ depends only on the isogeny class of E.

By expanding the product, it is clear that $L(E,s)$ is a Dirichlet series, i.e. of the form $\sum_{n\geq 1} a_n n^{-s}$ (this of course is the case for all zeta functions of varieties). We will set

$$f_E(\tau) = \sum_{n \geq 1} a_n q^n, \quad \text{where as usual } q = e^{2i\pi\tau}.$$

We can now state the first conjecture on L-functions of elliptic curves:

Conjecture 7.3.4. *The function $L(E,s)$ can be analytically continued to the whole complex plane to an entire function. Furthermore, there exists a positive integer N, such that if we set*

$$\Lambda(E,s) = N^{s/2}(2\pi)^{-s}\Gamma(s)L(E,s),$$

then we have the following functional equation:

$$\Lambda(E,2-s) = \pm\Lambda(E,s).$$

In this case, the Riemann hypothesis states that apart from the trivial zeros at non-positive integers, the zeros of $L(E,s)$ all lie on the critical line $\operatorname{Re} s = 1$.

The number N occurring in Conjecture 7.3.4 is a very important invariant of the curve. It is called the (analytic) conductor of E. From work of Carayol [Car], it follows that it must be equal to the (geometric) conductor of E which can be defined without reference to any conjectures. It suffices to say that it has the form $\prod_p p^{e_p}$, where the product is over primes of bad reduction, and for $p > 3$, $e_p = 1$ if E has multiplicative reduction at p, $e_p = 2$ if E has additive reduction. For $p \leq 3$, the recipe is more complicated and is given in Section 7.5.

One can also give a recipe for the $\pm$ sign occurring in the functional equation.

7.3.3 The Taniyama-Weil Conjecture

Now if the reader has a little acquaintance with modular forms, he will notice that the conjectured form of the functional equation of $L(E,s)$ is the same as the functional equation for the Mellin transform of a modular form of weight 2 over the group

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}), \quad c \equiv 0 \pmod{N} \right\}$$

(see [Lang4], [Ogg] or [Zag] for all relevant definitions about modular forms). Indeed, one can prove the following

Theorem 7.3.5. *Let f be a modular cusp form of weight 2 on the group $\Gamma_0(N)$ (equivalently $f\frac{dq}{q}$ is a differential of the first kind on $X_0(N) = \overline{\mathcal{H}/\Gamma_0(N)}$). Assume that f is a normalized newform (hence, in particular,*

an eigenform for the Hecke operators) and that f has rational Fourier coefficients. Then there exists an elliptic curve E defined over $\mathbb{Q}$ such that $f = f_E$, i.e. such that the Mellin transform of $f(it/\sqrt{N})$ is equal to $\Lambda(E,s)$.

Such a curve E is called a modular elliptic curve, and is a natural quotient of the Jacobian of the curve $X_0(N)$. Since analytic continuation and functional equations are trivial consequences of the modular invariance of modular forms we obtain:

Corollary 7.3.6. *Let E be a modular elliptic curve, and let $f = \sum_{n\geq 1} a_n q^n$ be the corresponding cusp form. Then Conjecture 7.3.4 is true for the curve E. In addition, it is known from Atkin-Lehner theory that one must have $f(-1/(N\tau)) = -\varepsilon N\tau^2 f(\tau)$ with $\varepsilon = \pm 1$. Then the functional equation is*

$$\Lambda(E, 2-s) = \varepsilon\Lambda(E,s).$$

(Please note the minus sign in the formula for $f(-1/(N\tau))$ which causes confusion and many mistakes in tables.) The number ε is called the sign of the functional equation.

With Theorem 7.3.5 in mind, it is natural to ask if the converse is true, i.e. whether every elliptic curve over $\mathbb{Q}$ is modular. This conjecture was first set forth by Taniyama. Its full importance and plausibility was understood only after Weil proved the following theorem, which we state only in an imprecise form (the precise statement can be found e.g. in [Ogg]):

Theorem 7.3.7 (Weil). *Let $f(\tau) = \sum_{n\geq 1} a_n q^n$, and for all primitive Dirichlet characters χ of conductor m set*

$$L(f,\chi,s) = \sum_{n\geq 1} \frac{a_n\chi(n)}{n^s},$$

$$\Lambda(f,\chi,s) = |Nm^2|^{s/2}(2\pi)^{-s}\Gamma(s)L(f,\chi,s).$$

Assume that these functions satisfy functional equations of the following form:

$$\Lambda(f,\chi,2-s) = w(\chi)\Lambda(f,\overline{\chi},s),$$

where $w(\chi)$ has modulus one, and assume that as χ varies, $w(\chi)$ satisfies certain compatibility conditions (being precise here would carry us a little too far). Then f is a modular form of weight 2 over $\Gamma_0(N)$.

Because of this theorem, the above conjecture becomes much more plausible. The Taniyama-Weil conjecture is then as follows:

Conjecture 7.3.8 (Taniyama-Weil). *Let E be an elliptic curve over $\mathbb{Q}$, let $L(E,s) = \sum_{n\geq 1} a_n n^{-s}$ be its L-series, and let $f_E(\tau) = \sum_{n\geq 1} a_n q^n$, so that*

the Mellin transform of $f_E(it/\sqrt{N})$ *is equal to* $\Lambda(E,s)$. *Then* f *is a cusp form of weight* 2 *on* $\Gamma_0(N)$ *which is an eigenfunction of the Hecke operators. Furthermore, there exists a morphism* ϕ *of curves from* $X_0(N)$ *to* E, *defined over* $\mathbb{Q}$, *such that the inverse image by* ϕ *of the differential* $dx/(2y+a_1x+a_3)$ *is the differential* $c(2i\pi)f(\tau)d\tau = cf(\tau)dq/q$, *where* c *is some constant.*

Note that the constant c, called Manin's constant, is conjectured to be always equal to ± 1 when ϕ is a "strong Weil parametrization" of E (see [Sil]).

A curve satisfying the Taniyama-Weil conjecture was called above a modular elliptic curve. Since this may lead to some confusion with modular curves (the curves $X_0(N)$) which are in general not elliptic, they are called Weil curves (which incidentally seems a little unfair to Taniyama).

The main theorem concerning this conjecture is Wiles's celebrated theorem, which states than when N is squarefree, the conjecture is true (see [Wiles], [Tay-Wil]). This result has been generalized by Diamond to the case where N is only assumed not to be divisible by 9 and 25. In addition, using Weil's Theorem 7.3.7, it was proved long ago by Shimura (see [Shi1] and [Shi2]) that it is true for elliptic curves with complex multiplication.

There is also a recent conjecture of Serre (see [Ser1]), which roughly states that any odd 2-dimensional representation of the Galois group $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ over a finite field must come from a modular form. It can be shown that Serre's conjecture implies the Taniyama-Weil conjecture.

The Taniyama-Weil conjecture, and hence the Taylor-Wiles proof, is mainly important for its own sake. However, it has attracted a lot of attention because of a deep result due to Ribet [Rib], saying that the Taniyama-Weil conjecture for squarefree N implies the full strength of Fermat's last "theorem" (FLT): if $x^n+y^n=z^n$ with x, y, z non-zero integers, then one must have $n \leq 2$. Thanks to Wiles, this is now really a theorem. Although it is not so interesting in itself, FLT has had amazing consequences on the development of number theory, since it is in large part responsible for the remarkable achievements of algebraic number theorists in the nineteenth century, and also as a further motivation for the study of elliptic curves, thanks to Ribet's result.

7.3.4 The Birch and Swinnerton-Dyer Conjecture

The other conjecture on elliptic curves which is of fundamental importance was stated by Birch and Swinnerton-Dyer after doing quite a lot of computer calculations on elliptic curves (see [Bir-SwD1], [Bir-SwD2]). For the remaining of this paragraph, we assume that we are dealing with a curve E defined over $\mathbb{Q}$ and satisfying Conjecture 7.3.4, for example a curve with complex multiplication, or more generally a Weil curve. (The initial computations of Birch and Swinnerton-Dyer were done on curves with complex multiplication).

Recall that we defined in a purely algebraic way the rank of an elliptic curve. A weak version of the Birch and Swinnerton-Dyer Conjecture (BSD) is that the rank is positive (i.e. $E(\mathbb{Q})$ is infinite) if and only if $L(E,1)=0$. This

is quite remarkable, and illustrates the fact that the function $L(E,s)$ which is obtained by putting together local data for every prime p, conjecturally gives information on global data, i.e. on the rational points.

The precise statement of the Birch and Swinnerton-Dyer conjecture is as follows:

Conjecture 7.3.9 (Birch and Swinnerton-Dyer). *Let E be an elliptic curve over $\mathbb{Q}$, and assume that Conjecture 7.3.4 (analytic continuation essentially) is true for E. Then if r is the rank of E, the function $L(E,s)$ has a zero of order exactly r at $s=1$, and in addition*

$$\lim_{s\to 1}(s-1)^{-r}L(E,s) = \Omega|Ш(E/\mathbb{Q})|R(E/\mathbb{Q})|E(\mathbb{Q})_{\mathrm{tors}}|^{-2}\prod_p c_p,$$

where Ω is a real period of E, $R(E/\mathbb{Q})$ is the so-called regulator of E, which is an $r\times r$ determinant formed by pairing in a suitable way a basis of the non-torsion points, the product is over the primes of bad reduction, c_p are small integers, and $Ш(E/\mathbb{Q})$ is the so-called Tate-Shafarevitch group of E.

It would carry us too far to explain in detail these quantities. Note only that the only quantity which is difficult to compute (in addition to the rank r) is the Tate-Shafarevitch group. In Sections 7.4 and 7.5 we will give algorithms to compute all the quantities which enter into this conjecture, except for $|Ш(E/\mathbb{Q})|$ which is then obtained by division (the result must be an integer, and in fact even a square, and this gives a check on the computations). More precisely, Section 7.5.3 gives algorithms for computing $\lim_{s\to 1}(s-1)^{-r}L(E,s)$, the quantities Ω and $|E(\mathbb{Q})_{\mathrm{tors}}|$ are computed using Algorithms 7.4.7 and 7.5.5, the regulator $R(E/\mathbb{Q})$ is obtained by computing a determinant of height pairings of a basis of the torsion-free part of $E(\mathbb{Q})$, these heights being computed using Algorithms 7.5.6 and 7.5.7. Finally, the c_p are obtained by using Algorithm 7.5.1 if $p\geq 5$ and Algorithm 7.5.2 if $p=2$ or 3.

Note that the above computational descriptions assume that we know a basis of the torsion-free part of $E(\mathbb{Q})$ and hence, in particular, the rank r, and that this is in general quite difficult.

The reader should compare Conjecture 7.3.9 with the corresponding result for the 0-dimensional case, i.e. Theorem 4.9.12. Dedekind's formula at $s=0$ is very similar to the BSD formula, with the regulator and torsion points playing the same role, and with the class group replaced by the Tate-Shafarevitch group, the units of K being of course analogous to the rational points.

Apart from numerous numerical verifications of BSD, few results have been obtained on BSD, and all are very deep. For example, only in 1987 was it proved by Rubin and Kolyvagin (see [Kol1], [Kol2], [Rub]) that Ш is finite for certain elliptic curves. The first result on BSD was obtained in 1977 by Coates and Wiles [Coa-Wil] who showed that if E has complex multiplication and if $E(\mathbb{Q})$ is infinite, then $L(E,1)=0$. Further results have been obtained,

in particular by Gross-Zagier, Rubin and Kolyvagin (see [Gro-Zag2], [GKZ], [Kol1], [Kol2]). For example, the following is now known:

Theorem 7.3.10. *Let E be a Weil curve. Then*

(1) *If $L(E,1) \neq 0$ then $r = 0$.*
(2) *If $L(E,1) = 0$ and $L'(E,1) \neq 0$ then $r = 1$.*

Furthermore, in both these cases $|Ш|$ is finite, and up to some simple factors divides the conjectural $|Ш|$ involved in BSD.

The present status of BSD is essentially that very little is known when the rank is greater than or equal to 2.

Another conjecture about the rank is that it is unbounded. This seems quite plausible. Using a construction of J.-F. Mestre (see [Mes3] and Exercise 9), Nagao has obtained an infinite family of curves of rank greater or equal to 13 (see [Nag]), and Mestre himself has just obtained an infinite family of curves of rank greater or equal to 14 (see [Mes5]). Furthermore, using Mestre's construction, several authors have obtained individual curves of much higher rank, the current record being rank 22 by Fermigier (see [Mes4], [Fer1], [Nag-Kou] and [Fer2]).

7.4 Algorithms for Elliptic Curves

The previous sections finish up our survey of results and conjectures about elliptic curves. Although the only results which we will need in what follows are the results giving the group law, and Theorems 7.2.14 and 7.2.15 giving basic properties of curves with complex multiplication, elliptic curves are a fascinating field of study *per se*, so we want to describe a number of algorithms to work on them. Most of the algorithms will be given without proof since this would carry us too far. Note that these algorithms are for the most part scattered in the literature, but others are part of the folklore or are new. I am particularly indebted to J.-F. Mestre and D. Bernardi for many of the algorithms of this section. The most detailed collection of algorithms on elliptic curves can be found in the recent book of Cremona [Cre].

7.4.1 Algorithms for Elliptic Curves over $\mathbb{C}$

The problems that we want to solve here are the following.

(1) Given ω_1 and ω_2, compute the coefficients g_2 and g_3 of the Weierstraß equation of the corresponding curve.
(2) Given ω_1 and ω_2 and a complex number z, compute $\wp(z)$ and $\wp'(z)$.
(3) Conversely given g_2 and g_3 such that $g_2^3 - 27g_3^2 \neq 0$, compute ω_1 and ω_2 (which are unique only up to an element of $\mathrm{SL}_2(\mathbb{Z})$).

(4) Similarly, given g_2, g_3 and a point (x, y) on the corresponding Weierstraß curve, compute the complex number z (unique up to addition of an element of the period lattice generated by ω_1 and ω_2) such that $x = \wp(z)$ and $y = \wp'(z)$.

If necessary, after exchanging ω_1 and ω_2, we may assume that $\mathrm{Im}(\omega_1/\omega_2) > 0$, i.e. if we set $\tau = \omega_1/\omega_2$ then $\tau \in \mathcal{H}$. As usual, we always set $q = e^{2i\pi\tau}$, and we have $|q| < 1$ when $\tau \in \mathcal{H}$. Then we have the following proposition:

Proposition 7.4.1. *We have*

$$g_2 = \frac{1}{12}\left(\frac{2\pi}{\omega_2}\right)^4\left(1 + 240\sum_{n\geq 1}\frac{n^3q^n}{1-q^n}\right)$$

and also

$$g_3 = \frac{1}{216}\left(\frac{2\pi}{\omega_2}\right)^6\left(1 - 504\sum_{n\geq 1}\frac{n^5q^n}{1-q^n}\right).$$

This could already be used to compute g_2 and g_3 reasonably efficiently, but it would be slow when τ is close to the real line. In this case, one should first find the complex number τ' belonging to the fundamental domain $\mathcal{F}$ which is equivalent to τ, compute g_2 and g_3 for τ', and then come back to τ using the (trivial) transformation laws of g_2 and g_3, i.e. $g_k(a\omega_1 + b\omega_2, c\omega_1 + d\omega_2) = g_k(\omega_1, \omega_2)$ when $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$. This leads to the following algorithms.

Algorithm 7.4.2 (Reduction). Given $\tau \in \mathcal{H}$, this algorithm outputs the unique τ' equivalent to τ under the action of $\mathrm{SL}_2(\mathbb{Z})$ and which belongs to the standard fundamental domain $\mathcal{F}$, as well as the matrix $A \in \mathrm{SL}_2(\mathbb{Z})$ such that $\tau' = A\tau$.

1. [Initialize] Set $A \leftarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.
2. [Reduce real part] Let $n \leftarrow \lfloor \mathrm{Re}(\tau) \rceil$, $\tau \leftarrow \tau - n$, $A \leftarrow \begin{pmatrix} 1 & -n \\ 0 & 1 \end{pmatrix} \cdot A$.
3. [Finished] Set $m \leftarrow \tau\overline{\tau}$. If $m \geq 1$, output τ and A and terminate the algorithm. Otherwise set $\tau \leftarrow -\overline{\tau}/m$, $A \leftarrow \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot A$ and go to step 2.

This is of course closely related to the reduction algorithm for positive definite quadratic forms (Algorithm 5.4.2), as well as to Gauss's lattice reduction algorithm in dimension 2 (Algorithm 1.3.14).

Warning. The condition $m \geq 1$ in step 3 should in practice be implemented as $m > 1 - \varepsilon$ for some $\varepsilon > 0$ depending on the current accuracy. If this precaution is not taken the algorithm may loop indefinitely, and the cost is simply that the final τ may land very close to but not exactly in the standard fundamental domain, and this has absolutely no consequence for practical computations.

We can now give the algorithm for computing g_2 and g_3.

Algorithm 7.4.3 (g_2 and g_3). Given ω_1 and ω_2 generating a lattice L, this algorithm computes the coefficients g_2 and g_3 of the Weierstraß equation of the elliptic curve $\mathbb{C}/L$.

1. [Initialize] If $\mathrm{Im}(\omega_1/\omega_2) < 0$, exchange ω_1 and ω_2. Then set $\tau \leftarrow \omega_1/\omega_2$.
2. [Reduce] Using Algorithm 7.4.2, find a matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$ such that $\tau' = A\tau$ is in the fundamental domain $\mathcal{F}$. Set $q' = e^{2i\pi\tau'}$.
3. [Compute] Compute g_2 and g_3 using the formulas given in Proposition 7.4.1, replacing q by q' and ω_2 by $c\omega_1 + d\omega_2$, and terminate the algorithm.

Since $\tau' \in \mathcal{F}$, we have $\mathrm{Im}\,\tau' \geq \sqrt{3}/2$ hence $|q| \leq e^{-\pi\sqrt{3}} \approx 4.33 \cdot 10^{-3}$, so the convergence of the series, although linear, will be very fast.

We can also use the power series expansions to compute $\wp(z)$ and $\wp'(z)$:

Proposition 7.4.4. *Set $\tau = \omega_1/\omega_2 \in \mathcal{H}$, $q = e^{2i\pi\tau}$ and $u = e^{2i\pi z/\omega_2}$. Then*

$$\wp(z) = \left(\frac{2i\pi}{\omega_2}\right)^2 \left(\frac{1}{12} + \frac{u}{(1-u)^2} + \sum_{n=1}^{\infty} q^n \left(u\left(\frac{1}{(1-q^n u)^2} + \frac{1}{(q^n-u)^2}\right) - \frac{2}{(1-q^n)^2}\right)\right)$$

and

$$\wp'(z) = \left(\frac{2i\pi}{\omega_2}\right)^3 u\left(\frac{1+u}{(1-u)^3} + \sum_{n=1}^{\infty} q^n \left(\frac{1+q^n u}{(1-q^n u)^3} + \frac{q^n+u}{(q^n-u)^3}\right)\right).$$

Note that the formula for $\wp'(z)$ in the first printing of [Sil] is incorrect.

As usual, we must do reductions of τ and z before applying the crude formulas, and this gives the following algorithm.

Algorithm 7.4.5 ($\wp(z)$ and $\wp'(z)$). Given ω_1 and ω_2 generating a lattice L, and $z \in \mathbb{C}$, this algorithm computes $\wp(z)$ and $\wp'(z)$.

1. [Initialize and reduce] If $\mathrm{Im}(\omega_1/\omega_2) < 0$, exchange ω_1 and ω_2. Then set $\tau \leftarrow \omega_1/\omega_2$. Using Algorithm 7.4.2, find a matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z})$ such that $A\tau$ is in the the fundamental domain $\mathcal{F}$. Finally, set $\tau \leftarrow A\tau$ and $\omega_2 \leftarrow c\omega_1 + d\omega_2$.
2. [Reduce z] Set $z \leftarrow z/\omega_2$, $n \leftarrow \lfloor \mathrm{Im}(z)/\mathrm{Im}(\tau) \rceil$, $z \leftarrow z - n\tau$ and $z \leftarrow z - \lfloor \mathrm{Re}(z) \rceil$.
3. [Compute] If $z = 0$, output a message saying that $z \in L$. Otherwise compute $\wp(z)$ and $\wp'(z)$ using the formulas given in Proposition 7.4.4 (with $u = e^{2i\pi z}$ since we have already divided z by ω_2) and terminate the algorithm.

Remark. For the above computations it is more efficient to use the formulas that link elliptic functions with the σ function, since the latter are theta series and so can be computed efficiently. For reasonable accuracy however (say less than 100 decimal digits) the above formulas suffice.

We now consider the inverse problems. Given g_2 and g_3 defining a Weierstraß equation, we want to compute a basis ω_1 and ω_2 of the corresponding lattice.

First, recall the definition of the Arithmetic-Geometric Mean (AGM) of two numbers.

Definition 7.4.6. *Let a and b be two positive real numbers. The Arithmetic-Geometric mean of a and b, denoted by $\mathrm{AGM}(a,b)$ is defined as the common limit of the two sequences a_n and b_n defined by $a_0 = a$, $b_0 = b$, $a_{n+1} = (a_n + b_n)/2$ and $b_{n+1} = \sqrt{a_n b_n}$.*

It is an easy exercise to show that these sequences converge and that they have a common limit $\mathrm{AGM}(a,b)$ (see Exercise 10). It can also be proved quite easily that

$$\frac{\pi}{2\,\mathrm{AGM}(a,b)} = \int_0^{\pi/2} \frac{dt}{\sqrt{a^2\cos^2 t + b^2 \sin^2 t}}$$

(see Exercise 11) and this can easily be transformed into an elliptic integral, which explains the relevance of the AGM to our problems. For many more details on the AGM, I refer to the marvelous book of Borwein and Borwein [Bor-Bor].

Apart from their relevance to elliptic integrals, the fundamental property of the AGM sequences a_n and b_n is that they converge quadratically, i.e. the number of significant decimals approximately *doubles* with each iteration (see Exercise 10). For example, there exists AGM-related methods for computing π to high precision (see again [Bor-Bor]), and since $2^{20} > 10^6$ only 20 iterations are needed to compute 1000000 decimals of π!

The AGM can also be considered when a and b are not positive real numbers but are arbitrary complex numbers. Here the situation is more complicated, but can be summarized as follows. At each stage of the iteration, we must choose some square root of $a_n b_n$. Assume that for n sufficiently large the same branch of the square root is taken (for example the principal branch, but it can be any other branch). Then the sequences again converge quadratically to the same limit, but this limit of course now depends on the choices made for the square roots. In addition, the set of values of $\pi/\mathrm{AGM}(a,b)$ (where now $\mathrm{AGM}(a,b)$ has infinitely many values) together with 0 form a *lattice* L in $\mathbb{C}$. The precise link with elliptic curves is as follows. Let e_1, e_2, e_3 be the three complex roots of the polynomial $4x^3 - g_2 x - g_3$ such that $y^2 = 4x^3 - g_2 x - g_3$ defines an elliptic curve E. Then, when the AGM runs through all its possible determinations $\pi/\mathrm{AGM}(\sqrt{e_1 - e_3}, \sqrt{e_1 - e_2})$ gives all the lattice points (except 0) of the lattice L such that $E \simeq \mathbb{C}/L$.

We however will usually use the AGM over the positive real numbers, where it is single-valued, since the elliptic curves that we will mainly consider are defined over $\mathbb{R}$, and even over $\mathbb{Q}$. In this case, the following algorithm gives a basis of the period lattice L. Since our curves will usually be given by a generalized Weierstraß equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ instead of the simpler equation $Y^2 = 4X^3 - g_2X - g_3$, we give the algorithm in that context.

Algorithm 7.4.7 (Periods of an Elliptic Curve over $\mathbb{R}$). Given real numbers $a_1, \ldots, a_6$, this algorithm computes the basis (ω_1, ω_2) of the period lattice of E such that ω_2 is a positive real number and ω_1/ω_2 has positive imaginary part and a real part equal to 0 or $-1/2$.

1. [Initialize] Using Formulas (7.1), compute b_2, b_4, b_6 and Δ, and if $\Delta < 0$ go to step 3.
2. [Disconnected case] Let e_1, e_2 and e_3 be the three real roots of the polynomial $4x^3 + b_2x^2 + 2b_4x + b_6 = 0$ with $e_1 > e_2 > e_3$. Set $\omega_2 \leftarrow \pi/\mathrm{AGM}(\sqrt{e_1 - e_3}, \sqrt{e_1 - e_2})$, $\omega_1 \leftarrow i\pi/\mathrm{AGM}(\sqrt{e_1 - e_3}, \sqrt{e_2 - e_3})$ and terminate the algorithm.
3. [Connected case] Let e_1 be the unique real root of $4x^3 + b_2x^2 + 2b_4x + b_6 = 0$. Set $a \leftarrow 3e_1 + b_2/4$ and $b \leftarrow \sqrt{3e_1^2 + (b_2/2)e_1 + b_4/2}$. Then set $\omega_2 \leftarrow 2\pi/\mathrm{AGM}(2\sqrt{b}, \sqrt{2b + a})$, $\omega_1 \leftarrow -\omega_2/2 + i\pi/\mathrm{AGM}(2\sqrt{b}, \sqrt{2b - a})$ and terminate the algorithm.

Note that the "real period" Ω occurring in the Birch and Swinnerton-Dyer conjecture 7.3.9 is $2\omega_2$ when $\Delta > 0$, and ω_2 otherwise, and that ω_1/ω_2 is not necessarily in the standard fundamental domain for $\mathcal{H}/\mathrm{SL}_2(\mathbb{Z})$.

Finally, we need an algorithm to compute the functional inverse of the $\wp$ function.

The Weierstraß parametrization $(\wp(z) : \wp'(z) : 1)$ can be seen as an exponential morphism from the universal covering $\mathbb{C}$ of $E(\mathbb{C})$. It can be considered as the composition of three maps:

$$\mathbb{C} \longrightarrow \mathbb{C}^* \longrightarrow \mathbb{C}^*/q^{\mathbb{Z}} \longrightarrow E(\mathbb{C})$$

$$z \mapsto u = e^{2i\pi z/\omega_2} \mapsto u \bmod q^{\mathbb{Z}} \mapsto (\wp(z), \wp'(z)),$$

the last one being an isomorphism. Its functional inverse, which we can naturally call the elliptic logarithm, is thus a multi-valued function. In fact, Algorithm 7.4.7 can be extended so as to find the inverse image of a given point. Since square roots occur, this give rise to the same indeterminacy as before, i.e. the point z is defined only up to addition of a point of the period lattice L. As in the previous algorithm, taking the positive square root in the real case gives directly the unique u such that $|q| < |u| \leq 1$. We will therefore only give the description for a real point.

Algorithm 7.4.8 (Elliptic Logarithm). Given real numbers $a_1, \ldots, a_6$ defining a generalized Weierstraß equation for an elliptic curve E and a point $P = (x, y)$ on $E(\mathbb{R})$, this algorithm computes the unique complex number z such that $\wp(z) = x + b_2/12$ and $\wp'(z) = 2y + a_1x + a_3$, where $\wp$ is the Weierstraß function corresponding to the period lattice of E, and which satisfies the following additional conditions. Either z is real and $0 \leq z < \omega_2$, or $\Delta > 0$, $z - \omega_1/2$ is real and satisfies $0 \leq z - \omega_1/2 < \omega_2$.

1. [Initialize] Using Formulas (7.1), compute b_2, b_4, b_6 and Δ. If $\Delta < 0$ go to step 6.
2. [Disconnected case] Let e_1, e_2 and e_3 be the three real roots of the polynomial $4x^3 + b_2x^2 + 2b_4x + b_6 = 0$ with $e_1 > e_2 > e_3$. Set $a \leftarrow \sqrt{e_1 - e_3}$ and $b \leftarrow \sqrt{e_1 - e_2}$. If $x < e_1$ set $f \leftarrow 1$, $\lambda \leftarrow y/(x - e_3)$ and $x \leftarrow \lambda^2 + a_1\lambda - a_2 - x - e_3$, otherwise set $f \leftarrow 0$. Finally, set $c \leftarrow \sqrt{x - e_3}$.
3. [Loop] Repeat $(a, b, c) \leftarrow ((a + b)/2, \sqrt{ab}, (c + \sqrt{c^2 + b^2 - a^2})/2)$ until the difference $a - b$ is sufficiently small.
4. [Connected component] If $f = 0$ and $2y + a_1x + a_3 < 0$ or $f = 1$ and $2y + a_1x + a_3 \geq 0$ set $z \leftarrow \arcsin(a/c)/a$. Otherwise set $z \leftarrow (\pi - \arcsin(a/c))/a$. If $f = 0$ output z and terminate the algorithm.
5. [Other component] Compute $\omega_1 \leftarrow i\pi/\operatorname{AGM}(\sqrt{e_1 - e_3}, \sqrt{e_2 - e_3})$ as in Algorithm 7.4.7 (unless of course this has already been done). Output $z + \omega_1/2$ and terminate the algorithm.
6. [Connected case] Let e_1 be the unique real root of $4x^3 + b_2x^2 + 2b_4x + b_6 = 0$. Set $\beta \leftarrow \sqrt{3e_1^2 + (b_2/2)e_1 + b_4/2}$, $\alpha \leftarrow 3e_1 + b_2/4$, $a \leftarrow 2\sqrt{\beta}$, $b \leftarrow \sqrt{\alpha + 2\beta}$ and $c \leftarrow (x - e_1 + \beta)/\sqrt{x - e_1}$.
7. [Loop] Repeat $(a, b, c) \leftarrow ((a + b)/2, \sqrt{ab}, (c + \sqrt{c^2 + b^2 - a^2})/2)$ until the difference $a - b$ is sufficiently small.
8. [Terminate] If $(2y + a_1x + a_3)((x - e_1)^2 - \beta^2) < 0$, set $z \leftarrow \arcsin(a/c)/a$ otherwise set $z \leftarrow (\pi - \arcsin(a/c))/a$. If $2y + a_1x + a_3 > 0$, set $z \leftarrow z + \pi/a$. Output z and terminate the algorithm.

Note that we could have avoided the extra AGM in step 5, but this would have necessitated using the complex AGM and arcsin. Hence, it is simpler to proceed as above. In addition, in practice ω_1 will have already been computed previously and so there is not really any extra AGM to compute.

7.4.2 Algorithm for Reducing a General Cubic

The problem that we want to solve here is the following. Given a general non-singular irreducible projective plane cubic over an arbitrary field K, say

$$s_1U^3 + s_2U^2V + s_3UV^2 + s_4V^3 + (s_5U^2 + s_6UV + s_7V^2)W + (s_8U + s_9V)W^2 + s_{10}W^3 \ ,$$

where $(U : V : W)$ are the projective coordinates, and a K-rational point $P_0 = (u_0 : v_0 : w_0)$ on the curve, find a birational transformation which transforms this into a generalized Weierstraß equation.

We will explain how to do this in the generic situation (i.e. assuming that no expression vanishes, that our points are in general position, etc ...), and then give the algorithm in general. We also assume for simplicity that our field is of characteristic different from 2.

We first make a couple of reductions. Since the curve is non-singular, its partial derivatives with respect to U and V cannot vanish simultaneously on the curve. Hence, by exchanging if necessary U and V, we may assume that it is the derivative with respect to V at P_0 which is different from zero. Consider now the tangent at P_0 to the curve. This tangent will then have a (rational) slope λ, and intersects the curve in a unique third point which we will call $P_1 = (u_1 : v_1 : w_1)$. After making the change of coordinates $(U', V') = (U - u_1, V - v_1)$ we may assume that P_1 has coordinates $(0 : 0 : 1)$, i.e. is at the origin, or in other words that the new value of s_{10} is equal to zero. We now have the following theorem (for simplicity we state everything with affine coordinates, but the conversion to projective coordinates is easy to make).

Theorem 7.4.9. *We keep the above notations and reductions. Call $c_j(U, V)$ the coefficient of degree W^{3-j} in the equation of the curve (so that c_j is a homogeneous polynomial of degree j), and set*

$$d(U, V) = c_2(U, V)^2 - 4c_1(U, V)c_3(U, V).$$

Furthermore, if λ is the slope of the tangent at P_0 as defined above, set

$$d(U, \lambda U + 1) = AU^4 + BU^3 + CU^2 + DU + E.$$

Then

(1) *We have $A = 0$ and $B \neq 0$.*

(2) *The transformation*

$$X = \frac{BU}{V - \lambda U}$$

$$Y = \frac{B}{(V - \lambda U)^2}(2c_3(U, V) + c_2(U, V))$$

is a birational transformation whose inverse is given by

$$U = X\frac{BY - c_2(X, \lambda X + B)}{2c_3(X, \lambda X + B)}$$

$$V = (\lambda X + B)\frac{BY - c_2(X, \lambda X + B)}{2c_3(X, \lambda X + B)}.$$

(3) *This birational map transforms the equation of the curve into the Weierstraß equation*

$$Y^2 = X^3 + CX^2 + BDX + B^2E.$$

Proof. The line $V = \lambda U$ is the new equation of the tangent at P_0 that we started from. This means that it is tangent to the curve. Solving for U, one has the trivial solution $U = 0$ corresponding to the point P_1, and the two other roots must be equal. In other words we must have $d(1, \lambda) = 0$, since this is the discriminant of the quadratic equation. Since clearly $A = d(1, \lambda)$, this shows that $A = 0$.

Now solving for the double root, we see that the coordinates of P_0 (in the new coordinate system of course) are $(\alpha, \lambda\alpha)$, where we set

$$\alpha = -\frac{c_2(1,\lambda)}{2c_3(1,\lambda)}.$$

Now I claim that we have the equalities

$$B = \frac{\partial d}{\partial V}(1,\lambda) = -4c_3(1,\lambda)\frac{\partial f}{\partial V}(\alpha,\lambda\alpha),$$

where $f(U, V) = 0$ is the (affine) equation of the curve. Assuming this for a moment, this last partial derivative is the partial derivative of f with respect to V at the point P_0, hence is different from zero by the first reduction made above. Furthermore, $c_3(1, \lambda) \neq 0$ also since otherwise P_0 would be at infinity and we have assumed (for the moment) that P_0 is in general position. This shows that $B \neq 0$ and hence the first part of the theorem. To prove my claim, note that the first equality is trivial. For the second, let us temporarily abbreviate $c_j(1, \lambda)$ to c_j and $\frac{\partial c_j}{\partial V}(1, \lambda)$ to c'_j. Then by homogeneity, one sees immediately that

$$\frac{\partial f}{\partial V}(\alpha,\lambda\alpha) = \frac{c_2^2c_3' - 2c_2c_3c_2' + 4c_1'c_3^2}{4c_3^2}.$$

We know that $A = c_2^2 - 4c_1c_3 = 0$ (and this can be checked once again explicitly if desired). Therefore we can replace c_2^2 by $4c_1c_3$, thus giving

$$\frac{\partial f}{\partial V}(\alpha,\lambda\alpha) = \frac{4c_1c_3' + 4c_1'c_3 - 2c_2c_2'}{4c_3}$$

and the claim follows by differentiating the formula $d = c_2^2 - 4c_1c_3$.

By simple replacement, one sees immediately that, since $B \neq 0$, the maps $(U, V) \to (X, Y)$ and $(X, Y) \to (U, V)$ are inverse to one another, hence the second part is clear.

For the last part, we simply replace U and V by their expressions in terms of X and Y. We can multiply by $c_3(X, \lambda X + B)$ (which is not identically zero), and we can also simplify the resulting equation by $BY - c_2(X, \lambda X + B)$ since B is different from zero and the curve is irreducible (why?). After expanding and simplifying we obtain the equation

$$B^2Y^2 = d(X, \lambda X + B).$$

Now since $d(U, V)$ is a homogeneous polynomial of degree 4, one sees immediately that

$$d(X, \lambda X + B) = B^2X^3 + CB^2X^2 + DB^3X + EB^4,$$

thus finishing the proof of the theorem. □

It is now easy to generalize this theorem to the case where the point P_0 is not in general position, and this leads to the following algorithm, which we give in projective coordinates.

Algorithm 7.4.10 (Reduction of a General Cubic). Let K be a field of characteristic different from 2, and let $f(U, V, W) = 0$ be the equation of a general cubic, where

$$f(U,V,W) = s_1U^3 + s_2U^2V + s_3UV^2 + s_4V^3 + (s_5U^2 + s_6UV + s_7V^2)W + (s_8U + s_9V)W^2 + s_{10}W^3.$$

Finally, let $P_0 = (u_0 : v_0 : w_0)$ be a point on the cubic, i.e. such that $f(u_0, v_0, w_0) = 0$. This algorithm, either outputs a message saying that the curve is singular or reducible, or else gives a Weierstraß equation for the curve and a pair of inverse birational maps which transform one equation into the other. We will call $(X : Y : T)$ the new projective coordinates, and continue to call s_i the coefficients of the transformed equation g during the algorithm.

1. [Initialize] Set $(m_1, m_2, m_3) \leftarrow (U, V, W)$, $(n_1, n_2, n_3) \leftarrow (X, Y, T)$ and $g \leftarrow f$. (Here $(m_1 : m_2 : m_3)(U, V, W)$ and $(n_1 : n_2 : n_3)(X, Y, T)$ will be the pair of inverse birational maps. The assignments given in this algorithm for these maps and for g are formal, i.e. we assign polynomials or rational functions, not values. In addition, it is understood that the modifications of g imply the modifications of the coefficients s_i.)
2. [Send P_0 to $(0 : 0 : 1)$] If $w_0 \neq 0$, set $(m_1, m_2, m_3) \leftarrow (w_0m_1 - u_0m_3, w_0m_2 - v_0m_3, w_0m_3)$, $(n_1, n_2, n_3) \leftarrow (w_0n_1 + u_0n_3, w_0n_2 + v_0n_3, w_0n_3)$, $g \leftarrow g(w_0U + u_0W, w_0V + v_0W, w_0W)$ and go to step 3. Otherwise, if $u_0 \neq 0$, set $(m_1, m_2, m_3) \leftarrow (u_0m_3, u_0m_2 - v_0m_1, u_0m_1)$, $(n_1, n_2, n_3) \leftarrow (u_0n_3, u_0n_2 + v_0n_3, u_0n_1)$, $g \leftarrow g(u_0W, u_0V + v_0W, u_0U)$ and go to step 3. Finally, if $w_0 = u_0 = 0$ (hence $v_0 \neq 0$), exchange m_2 and m_3, n_2 and n_3, and set $g \leftarrow g(U, W, V)$.
3. [Exchange U and V?] (Here $s_{10} = 0$). If $s_8 = s_9 = 0$, output a message saying that the curve is singular at P_0 and terminate the algorithm. Otherwise, if $s_9 = 0$, exchange m_1 and m_2, n_1 and n_2, and set $g \leftarrow g(V, U, W)$.
4. [Send P_1 to $(0 : 0 : 1)$] (Here $s_9 \neq 0$.) Set $\lambda \leftarrow (-s_8/s_9)$, $c_2 \leftarrow s_7\lambda^2 + s_6\lambda + s_5$, $c_3 \leftarrow s_4\lambda^3 + s_3\lambda^2 + s_2\lambda + s_1$. Then, if $c_3 \neq 0$, set $(m_1, m_2, m_3) \leftarrow (c_3m_1 + c_2m_3, c_3m_2 + \lambda c_2m_3, c_3m_3)$, $(n_1, n_2, n_3) \leftarrow (c_3n_1 - c_2n_3, c_3n_2 - \lambda c_2n_3, c_3n_3)$,

$g \leftarrow g(c_3U - c_2W, c_3V - \lambda c_2W, c_3W)$ and go to step 5. Otherwise, if $c_2 = 0$ output a message saying that the curve is reducible and terminate the algorithm. Finally, if $c_3 = 0$ and $c_2 \neq 0$, set $(m_1, m_2, m_3) \leftarrow (m_3, m_2 - \lambda m_1, m_1)$, $(n_1, n_2, n_3) \leftarrow (n_3, n_2 + \lambda n_3, n_1)$ and $g \leftarrow g(W, V + \lambda W, U)$, then set $\lambda \leftarrow 0$.

5. [Apply theorem] (Here we are finally in the situation of the theorem.) Let as in the theorem $c_j(U,V)$ be the coefficient of W^{3-j} in $g(U,V,W)$, and $d(U,V) \leftarrow c_2(U,V)^2 - 4c_1(U,V)c_3(U,V)$. Compute B, C, D and E such that $d(U, \lambda U + 1) = BU^3 + CU^2 + DU + E$. Then set

$$(m_1, m_2, m_3) \leftarrow (Bm_1(m_2 - \lambda m_1)m_3, \\ B(2c_3(m_1, m_2) + c_2(m_1, m_2)m_3), (m_2 - \lambda m_1)^2 m_3),$$

$$(n_1, n_2, n_3) \leftarrow (n_1(Bn_2n_3 - c_2(n_1, \lambda n_1 + Bn_3)), \\ (\lambda n_1 + Bn_3)(Bn_2n_3 - c_2(n_1, \lambda n_1 + Bn_3)), 2c_3(n_1, \lambda n_1 + Bn_3)).$$

Output the maps $(X, Y, T) \leftarrow (m_1, m_2, m_3)$ and $(U, V, W) \leftarrow (n_1, n_2, n_3)$, the projective Weierstraß equation

$$Y^2T = X^3 + CX^2T + DBXT^2 + EB^2T^3$$

and terminate the algorithm.

7.4.3 Algorithms for Elliptic Curves over $\mathbb{F}_p$

The only algorithms which we will need here are algorithms which count the number of points of an elliptic curve over $\mathbb{F}_p$, or equivalently the numbers a_p such that $|E(\mathbb{F}_p)| = p + 1 - a_p$. We first describe the naïve algorithm which expresses a_p as a sum of Legendre symbols, then give a much faster algorithm using Shanks's baby-step giant-step method and a trick of Mestre.

Counting the number of points over $\mathbb{F}_2$ or $\mathbb{F}_3$ is trivial, so we assume that $p \geq 5$. In particular, we may simplify the Weierstraß equation, i.e. assume that $a_1 = a_2 = a_3 = 0$, so the equation of the curve is of the form $y^2 = x^3 + ax + b$. The curve has one point at infinity $(0:1:0)$, and then for every $x \in \mathbb{F}_p$, there are $1 + \left(\frac{x^3+ax+b}{p}\right)$ values of y. Hence we have $N_p = p + 1 + \sum_{x \in \mathbb{F}_p} \left(\frac{x^3+ax+b}{p}\right)$, thus giving the formula

$$a_p = -\sum_{x \in \mathbb{F}_p} \left(\frac{x^3 + ax + b}{p}\right).$$

This formula gives a $O(p^{1+o(1)})$ time algorithm for computing a_p, and this is reasonable when p does not exceed 10000, say.

However we can use Shanks's baby step-giant step method to obtain a much better algorithm. By Hasse's theorem, we know that $p + 1 - 2\sqrt{p} <$

$N_p < p+1+2\sqrt{p}$, hence we can apply Algorithm 5.4.1 with $C = p+1-2\sqrt{p}$ and $B = p+1+2\sqrt{p}$. This will give an algorithm which runs in time $(B-C)^{1/2+o(1)} = p^{1/4+o(1)}$, and so will be much faster for large p. Now the reader will recall that one problem with Shanks's method is that if our group is not cyclic, or if we do not start with a generator of the group, we need to do some extra work which is not so easy to implement. There is a nice trick due, I believe to Mestre, which tells us how to do this extra work in a very simple manner.

If one considers all the curves over $\mathbb{F}_p$ defined by $y^2 = x^3 + ad^2x + bd^3$ with $d \neq 0$, then there are exactly two isomorphism classes of such curves: those for which $\left(\frac{d}{p}\right) = 1$ are all isomorphic to the initial curve correspond to $d = 1$, and those for which $\left(\frac{d}{p}\right) = -1$ are also all isomorphic, but to another curve. Call E' one of these other curves. Then one has the following proposition.

Proposition 7.4.11. *Let*

$$E(\mathbb{F}_p) \simeq \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \quad \text{and} \quad E'(\mathbb{F}_p) \simeq \mathbb{Z}/d_1'\mathbb{Z} \times \mathbb{Z}/d_2'\mathbb{Z}$$

be the Abelian group structures of $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_p)$ respectively, with $d_1 \mid d_2$ and $d_1' \mid d_2'$ (see Proposition 7.1.9). Then for $p > 457$ we have

$$\max(d_2, d_2') > 4\sqrt{p} \ .$$

This proposition shows that on at least one of the two curves E or E' there will be points of order greater than $4\sqrt{p}$, hence according to Hasse's theorem, sufficiently large so as to obtain the cardinality of $E(\mathbb{F}_p)$ (or of $E'(\mathbb{F}_p)$) immediately using Shanks's baby-step giant-step method. In addition, since each value of x gives either two points on one of the curves and none on the other, or one on each, it is clear that if $|E(\mathbb{F}_p)| = p+1-a_p$, we have $|E'(\mathbb{F}_p)| = p+1+a_p$, so computing one value gives immediately the other one.

This leads to the following algorithm.

Algorithm 7.4.12 (Shanks-Mestre). Given an elliptic curve E over $\mathbb{F}_p$ with $p > 457$ by a Weierstraß equation $y^2 = x^3 + ax + b$, this algorithm computes the a_p such that $|E(\mathbb{F}_p)| = p+1-a_p$.

1. [Initialize] Set $x \leftarrow -1$, $A \leftarrow 0$, $B \leftarrow 1$, $k_1 = 0$.
2. [Get next point] (Here we have $|E(\mathbb{F}_p)| \equiv A \pmod{B}$). Repeat $x \leftarrow x+1$, $d \leftarrow x^3 + ax + b$, $k \leftarrow \left(\frac{d}{p}\right)$ until $k \neq 0$ and $k \neq k_1$. Set $k_1 \leftarrow k$. Finally, if $k_1 = -1$ set $A_1 \leftarrow 2p+2-A \bmod B$ else set $A_1 \leftarrow A$.
3. [Find multiple of the order of a point] Let m be the smallest integer such that $m > p+1-2\sqrt{p}$ and $m \equiv A_1 \pmod{B}$. Using Shanks's baby-step giant-step strategy, find an integer n such that $m \leq n < p+1+2\sqrt{p}$, $n \equiv m \pmod{B}$

and such that $n \cdot (xd, d^2) = 0$ on the curve $Y^2 = X^3 + ad^2X + bd^3$ (note that this will be isomorphic to the curve E or E' according to the sign of k_1).

4. [Find order] Factor n, and deduce from this the exact order h of the point (xd, d^2).

5. [Finished?] Using for instance the Chinese remainder algorithm, find the smallest integer h' which is a multiple of h and such that $h' \equiv A_1 \pmod{B}$. If $h' < 4\sqrt{p}$ set $B \leftarrow LCM(B, h)$, then $A \leftarrow h' \bmod B$ if $k_1 = 1$, $A \leftarrow 2p + 2 - h' \bmod B$ if $k_1 = -1$, and go to step 2.

6. [Compute a_p] Let N be the unique multiple of h' such that $p + 1 - 2\sqrt{p} < N < p + 1 + 2\sqrt{p}$. Output $a_p = p + 1 - k_1 N$ and terminate the algorithm.

The running time of this algorithm is $O(p^{1/4+\epsilon})$ for any $\epsilon > 0$, but it is much easier to implement than the algorithm for class numbers because of the simpler group structure. It should be used instead of the algorithm using Legendre symbols as soon as p is greater than 457. Note that one can prove that 457 is best possible, but it is easy to modify slightly the algorithm so that it works for much lower values of p.

Note also that, as in the case of class groups of quadratic fields, we can use the fact that the inverse of a point is trivial to compute, and hence enlarge by a factor $\sqrt{2}$ the size of the giant steps. In other words, in step 3 the size of the giant steps should be taken equal to the integer part of $\sqrt{2\sqrt{p}/B}$.

Another algorithm for computing a_p has been discovered by R. Schoof ([Scho]). What is remarkable about it is that it is a *polynomial time* algorithm, more precisely it runs in time $O(\ln^8 p)$. The initial version did not seem to be very useful in practice, but a lot of progress has been done since.

Schoof's idea, which we will not explain in detail here, is to use the *division polynomials* for the Weierstraß $\wp$ function, i.e. polynomials which express $\wp(nz)$ and $\wp'(nz)$ in terms of $\wp(z)$ and $\wp'(z)$ for integer n (in fact a prime number n). This gives *congruences* for the a_p, and using the Chinese remainder theorem we can glue together these congruences to compute the a_p.

An interesting blend of the baby-step giant-step algorithm and Schoof's algorithm is to compute Schoof-type congruences for a_p modulo a few primes ℓ. If for example we find the congruences modulo 2, 3 and 5, we can divide the search interval by 30 in the algorithm above, and hence this allows the treatment of larger primes.

The main practical problem with Schoof's idea is that the equations giving the division polynomials are of degree $(n^2 - 1)/2$, and this becomes very difficult to handle as soon as n is a little large.

Recently N. Elkies has been able to show that for approximately one half of the primes n, this degree can be reduced to $n + 1$, which is much more manageable. J.-M. Couveignes has also shown how to use n which are powers of small primes and not only primes.

Combining all these ideas, Morain and Lercier (Internet announcement) have been able to deal with a 500-digit prime, which is the current record at the time of this writing.

7.5 Algorithms for Elliptic Curves over $\mathbb{Q}$

7.5.1 Tate's algorithm

Given an elliptic curve E defined over $\mathbb{Q}$, using Algorithm 7.4.10 we can assume that E is given by a generalized Weierstraß equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ with coefficients in $\mathbb{Q}$. We would first like to find a global *minimal* Weierstraß equation of E (see [Sil], [LN476] and Algorithm 7.5.3 for the precise definitions). This will be a canonical way of representing the curve E since this equation exists and is unique. (As already remarked, it is essential at this point that we work over $\mathbb{Q}$ and not over an arbitrary number field.) Note that this is a major difference with the case of equations defining number fields, where no really canonical equation for the field can be found, but only partial approaches such as the pseudo-canonical polynomial given by Algorithm 4.4.12. In addition, it is necessary to know this minimal equation for several other algorithms.

Two elliptic curves with different parameters may be isomorphic over $\mathbb{Q}$. Such an isomorphism must be given by transformations $x = u^2x' + r, \quad y = u^3y' + su^2x' + t$, where $u \in \mathbb{Q}^*$, $r, s, t \in \mathbb{Q}$. We obtain a new model for the same elliptic curve. Using the same quantities as those used in Formulas (7.1), the parameters of the new model are given by

$$\begin{aligned}
ua'_1 &= a_1 + 2s, \quad u^2a'_2 = a_2 - sa_1 + 3r - s^2\\
u^3a'_3 &= a_3 + ra_1 + 2t\\
u^4a'_4 &= a_4 - sa_3 + 2ra_2 - (t + rs)a_1 + 3r^2 - 2st\\
u^6a'_6 &= a_6 + ra_4 + r^2a_2 + r^3 - ta_3 - t^2 - rta_1\\
u^2b'_2 &= b_2 + 12r, \quad u^4b'_4 = b_4 + rb_2 + 6r^2\\
u^6b'_6 &= b_6 + 2rb_4 + r^2b_2 + 4r^3\\
u^8b'_8 &= b_8 + 3rb_6 + 3r^2b_4 + r^3b_2 + 3r^4\\
u^4c'_4 &= c_4, \quad u^6c'_6 = c_6, \quad u^{12}\Delta' = \Delta, \quad j' = j, \quad u^{-1}\omega' = \omega.
\end{aligned} \tag{7.2}$$

Using these formulas, we may now assume that the coefficients of the equations are integers. We will make this assumption from now on. We first want to find a model for E which is minimal with respect to a given prime p, and we also want to know the type of the fiber at p of the elliptic pencil defined by E over $\mathbb{Z}$ (see [Sil], [LN476]). The possible types are described by symbols known as Kodaira types. They are $I_0, I_\nu, II, III, IV, I_0^*, I_\nu^*, II^*, III^*, IV^*$, where ν

is a positive integer. We need also to compute the coefficient c_p which appears in the formulation of the Birch and Swinnerton-Dyer Conjecture 7.3.9, that is, the index in $E(\mathbb{Q}_p)$ of the group $E^0(\mathbb{Q}_p)$ of points which do not reduce to the singular point.

The following algorithm is due to Tate (cf [LN476]). We specialize his description to the case of rational integers. The situation is a bit simpler when the prime p is greater than 3, so let us start with that case.

Algorithm 7.5.1 (Reduction of an Elliptic Curve Modulo p). Given integers $a_1, \ldots, a_6$ and a prime $p > 3$, this algorithm determines the Kodaira symbol associated with the curve modulo p. In addition, it computes the exponent f of p in the arithmetic conductor of the curve, the index $c = [E(\mathbb{Q}_p) : E^0(\mathbb{Q}_p]$ and integers u, r, s, t such that $a'_1, \ldots, a'_6$ linked to $a_1, \ldots, a_6$ via Formulas (7.2) give a model with the smallest possible power of p in its discriminant.

1. [Initialize] Compute c_4, c_6, Δ and j using Formulas (7.1). If $v_p(j) < 0$ set $k \leftarrow v_p(\Delta) + v_p(j)$ else set $k \leftarrow v_p(\Delta)$.
2. [Minimal?] If $k < 12$ set $u \leftarrow 1$, $r \leftarrow 0$, $s \leftarrow 0$, and $t \leftarrow 0$. Otherwise, set $u \leftarrow p^{\lfloor k/12 \rfloor}$; if a_1 is odd then set $s \leftarrow (u - a_1)/2$ else set $s \leftarrow -a_1/2$. Set $a'_2 \leftarrow a_2 - sa_1 - s^2$. Set $r \leftarrow -a'_2/3, (u^2 - a'_2)/3$ or $(-u^2 - a'_2)/3$ depending on a'_2 being congruent to 0, 1 or -1 modulo 3. Set $a'_3 \leftarrow a_3 + ra_1$. If a'_3 is odd, then set $t \leftarrow (u^3 - a'_3)/2$ else set $t \leftarrow -a'_3/2$. Finally, set $k \leftarrow k \bmod 12$, $\Delta \leftarrow \Delta/u^{12}$, $c_4 \leftarrow c_4/u^4$ and $c_6 \leftarrow c_6/u^6$.
3. [Non-integral invariant] If $v_p(j) < 0$, then set $\nu \leftarrow -v_p(j)$. k must be equal to 0 or 6. If $k = 0$, set $f \leftarrow 1$, and set $c \leftarrow \nu$ if $\left(\frac{-c_6}{p}\right) = 1$ or $c \leftarrow \gcd(2, \nu)$ if $\left(\frac{-c_6}{p}\right) = -1$, then output Kodaira type I_ν. If $k = 6$ set $f \leftarrow 2$, and set $c \leftarrow 3 + \left(\frac{\Delta c_6 p^{-9-\nu}}{p}\right)$ if ν is odd, $c \leftarrow 3 + \left(\frac{\Delta p^{-6-\nu}}{p}\right)$ if ν is even, then output Kodaira type I^*_ν. In any case, output f, c, u, r, s, t and terminate the algorithm.
4. [Integral invariant] If $k = 0$ then set $f \leftarrow 0$ else set $f \leftarrow 2$. The possible values for k are 0, 2, 3, 4, 6, 8, 9 and 10. Set $c \leftarrow 1, 1, 2, 2 + \left(\frac{-6c_6 p^{-2}}{p}\right), 1+$ the number of roots of $4X^3 - 3c_4 p^{-2} X - c_6 p^{-3}$ in $\mathbb{Z}/p\mathbb{Z}$, $2 + \left(\frac{-6c_6 p^{-4}}{p}\right), 2, 1$ respectively. Output respectively the Kodaira types $I_0, II, III, IV, I^*_0, IV^*, III^*, II^*$. In any case, output f, c, u, r, s, t and terminate the algorithm.

When $p = 2$ or $p = 3$, the algorithm is much more complicated.

Algorithm 7.5.2 (Reduction of an Elliptic Curve Modulo 2 or 3). Given integers $a_1, \ldots, a_6$ and $p = 2$ or 3, this algorithm determines the Kodaira symbol associated with the curve modulo p. In addition, it computes the exponent f of p in the arithmetic conductor of the curve, the index $c = [E(\mathbb{Q}_p) : E^0(\mathbb{Q}_p]$ and integers u, r, s, t such that $a'_1, \ldots, a'_6$ linked to $a_1, \ldots, a_6$ via Formulas (7.2) give a model with the smallest possible power of p in its discriminant. To simplify the presentation, we use a variable T which will hold the Kodaira type, coded in any way one likes.

1. [Initialize] Set $u \leftarrow 1$, $r \leftarrow 0$, $s \leftarrow 0$, and $t \leftarrow 0$. Compute Δ and j using Formulas (7.1). Set $\nu \leftarrow v_p(\Delta)$.

2. [Type I_0] If $\nu = 0$ then set $f \leftarrow 0$, $c \leftarrow 1$, $T \leftarrow I_0$ and go to step 22.

3. [Type I_ν] If $p \nmid b_2 = a_1^2 + 4a_2$ then set $f \leftarrow 1$, and set $c \leftarrow \nu$ if $X^2 + a_1 X - a_2$ has a root in $\mathbb{Z}/p\mathbb{Z}$, set $c \leftarrow \gcd(2, \nu)$ otherwise, then set $T \leftarrow I_\nu$ and go to step 22.

4. [Change Equation] If $p = 2$, then set $r_1 \leftarrow a_4 \bmod 2$, $s_1 \leftarrow (r_1 + a_2) \bmod 2$ and $t_1 \leftarrow (a_6 + r_1(a_4 + s_1)) \bmod 2$, otherwise compute b_6 using Formulas (7.1) and set $r_1 \leftarrow -b_6 \bmod 3$, $s_1 \leftarrow a_1 \bmod 3$ and $t_1 \leftarrow (a_3 + r_1 a_1) \bmod 3$. Use Formulas (7.2) with the parameters $1, r_1, s_1, t_1$ to compute $a_1', \ldots, a_6'$, then set $a_1 \leftarrow a_1'$, $a_2 \leftarrow a_2'$, $\ldots$, $a_6 \leftarrow a_6'$, $r \leftarrow r + u^2 r_1$, $s \leftarrow s + u s_1$ and $t \leftarrow t + u^3 t_1 + u^2 s r_1$.

5. [Type II] If $p^2 \nmid a_6$, then set $f \leftarrow \nu$, $c \leftarrow 1$, $T \leftarrow II$ and go to step 22.

6. [Type III] Compute b_8 using Formulas (7.1). If $p^3 \nmid b_8$, then set $f \leftarrow \nu - 1$, $c \leftarrow 2$, $T \leftarrow III$ and go to step 22.

7. [Type IV] Compute b_6 using Formulas (7.1). If $p^3 \nmid b_6$, then set $f \leftarrow \nu - 2$ and set $c \leftarrow 3$ if $X^2 + a_3/pX - a_6/p^2$ has a root in $\mathbb{Z}/p\mathbb{Z}$, set $c \leftarrow 1$ otherwise, then set $T \leftarrow IV$ and go to step 22.

8. [Change Equation] If $p^3 \nmid a_6$ do the following. If $p = 2$, then set $k \leftarrow 2$, otherwise set $k \leftarrow a_3 \bmod 9$. Use Formulas (7.2) with parameters $1, 0, 0, k$ to compute $a_1', \ldots, a_6'$, then set $a_1 \leftarrow a_1'$, $a_2 \leftarrow a_2'$, $\ldots$, $a_6 \leftarrow a_6'$ and finally set $t \leftarrow t + u^3 k$.

9. [Type I_0^*] (At this point, we have $p \mid a_2$, $p^2 \mid a_4$ and $p^3 \mid a_6$.) Set $P \leftarrow X^3 + a_2/pX^2 + a_4/p^2 X + a_6/p^3$. If P has distinct roots modulo p, then set $f \leftarrow \nu - 4$, set $c \leftarrow 1+$ the number of roots of P in $\mathbb{Z}/p\mathbb{Z}$, $T \leftarrow I_0^*$ and go to step 22.

10. [Change Equation] Let a be the multiple root of the polynomial P modulo p. If $a \neq 0$, then use Formulas (7.2) with parameters $1, ap, 0, 0$ to compute $a_1', \ldots, a_6'$, then set $a_1 \leftarrow a_1'$, $a_2 \leftarrow a_2'$, $\ldots$, $a_6 \leftarrow a_6'$, $r \leftarrow r + u^2 ap$ and $t \leftarrow t + u^2 sap$. If a is a double root, then go to step 16.

11. [Type IV^*] (Here $p^2 \mid a_3$, $p^4 \mid a_6$.) Set $P \leftarrow X^2 + a_3/p^2 X - a_6/p^4$. If P has a double root in $\mathbb{Z}/p\mathbb{Z}$, then let a be that root. Otherwise set $f \leftarrow \nu - 6$, set $c \leftarrow 3$ if P splits over $\mathbb{Z}/p\mathbb{Z}$ and $c \leftarrow 1$ otherwise, set $T \leftarrow IV^*$ and go to step 22.

12. [Change Equation] If $a \neq 0$ then use Formulas (7.2) with parameters $1, 0, 0, ap^2$ to compute $a_1', \ldots, a_6'$, then set $a_1 \leftarrow a_1'$, $a_2 \leftarrow a_2'$, $\ldots$, $a_6 \leftarrow a_6'$ and $t \leftarrow t + u^3 ap^2$.

13. [Type III^*] If $p^4 \nmid a_4$, then set $f \leftarrow \nu - 7$, $c \leftarrow 2$, $T \leftarrow III^*$ and go to step 22.

14. [Type II^*] If $p^6 \nmid a_6$, then set $f \leftarrow \nu - 8$, $c \leftarrow 1$, $T \leftarrow II^*$ and go to step 22.

15. [Non-minimal equation] Use Formulas (7.2) with parameters $p, 0, 0, 0$ to compute $a'_1, \ldots, a'_6$, then set $a_1 \leftarrow a'_1$, $a_2 \leftarrow a'_2$, $\ldots$, $a_6 \leftarrow a'_6$, $u \leftarrow pu$, $\nu \leftarrow \nu - 12$ and go to step 2.

16. [Initialize Loop] Set $f \leftarrow \nu - 5$, $\nu \leftarrow 1$, $q \leftarrow p^2$.

17. [Type I^*_ν, day in] Set $P \leftarrow X^2 + a_3/qX - a_6/q^2$. If P has distinct roots modulo p, then set $c \leftarrow 4$ if these roots are in $\mathbb{Z}/p\mathbb{Z}$, set $c \leftarrow 2$ otherwise, then set $T \leftarrow I^*_\nu$ and go to step 22.

18. [Change Equation] Let a be the double root of P modulo p. If $a \neq 0$, use Formulas (7.2) with parameters $1, 0, 0, aq$ to compute $a'_1, \ldots, a'_6$, then set $a_1 \leftarrow a'_1$, $a_2 \leftarrow a'_2$, $\ldots$, $a_6 \leftarrow a'_6$ and $t \leftarrow t + u^3 aq$.

19. [Type I^*_ν, day out] Set $\nu \leftarrow \nu + 1$ and $P \leftarrow a_2/pX^2 + a_4/(pq)X + a_6/(pq^2)$. If P has distinct roots modulo p, then set $c \leftarrow 4$ if these roots are in $\mathbb{Z}/p\mathbb{Z}$, set $c \leftarrow 2$ otherwise, then set $T \leftarrow I^*_\nu$ and go to step 22.

20. [Change Equation] Let a be the double root of P modulo p. If $a \neq 0$, use Formulas (7.2) with parameters $1, aq, 0, 0$ to compute $a'_1, \ldots, a'_6$, then set $a_1 \leftarrow a'_1$, $a_2 \leftarrow a'_2$, $\ldots$, $a_6 \leftarrow a'_6$, $r \leftarrow r + u^2 aq$ and $t \leftarrow t + u^2 saq$.

21. [Loop] Set $\nu \leftarrow \nu + 1$, $q \leftarrow p \cdot q$ and go to step 17.

22. [Common termination] Output the Kodaira type T, the numbers f, c, u, r, s, t and terminate the algorithm.

Let us turn now to the global counterpart of this process: what is the best equation for an elliptic curve defined over $\mathbb{Q}$?.

Algorithm 7.5.3 (Global Reduction of an Elliptic Curve). Given $a_1, \ldots, a_6 \in \mathbb{Z}$, this algorithm computes the arithmetic conductor N of the curve and integers u, r, s, t such that $a'_1, \ldots, a'_6$ linked to $a_1, \ldots, a_6$ via Formulas (7.2) give a model with the smallest possible discriminant (in absolute value) and such that $a'_1, a'_3 \in \{0, 1\}$ and $a'_2 \in \{0, \pm 1\}$.

1. [Initialize] Set $N \leftarrow 1$, $u \leftarrow 1$, $r \leftarrow 0$, $s \leftarrow 0$ and $t \leftarrow 0$. Compute $D \leftarrow |\Delta|$ using Formulas (7.1).

2. [Finished ?] If $D = 1$, then output N, u, r, s, t and terminate the algorithm.

3. [Local Reduction] Find a prime divisor p of D. Then use Algorithm 7.5.1 or 7.5.2 to compute the quantities f_p, u_p, r_p, s_p (the quantity c_p may be discarded if it is not wanted for other purposes). Set $N \leftarrow Np^{f_p}$. If $u_p \neq 1$, set $u \leftarrow uu_p$, $r \leftarrow r + u^2 r_p$, $s \leftarrow s + us_p$ and $t \leftarrow t + u^3 t_p + u^2 sr_p$. Finally, set $D \leftarrow D/p$ until $p \nmid D$, then go to step 2.

Note that if only the minimal Weierstraß equation of the curve is desired, and not all the local data as well, we can use a simpler algorithm due to Laska (see [Las] and Section 3.2 of [Cre] for a version due to Kraus and Connell).

7.5.2 Computing rational points

We now turn to the problem of trying to determine the group $E(\mathbb{Q})$ of rational points on E. As already mentioned, this is a difficult problem for which no algorithm exists unless we assume some of the standard conjectures.

On the other hand, the determination of the torsion subgroup $E(\mathbb{Q})_{\text{tors}}$ is easy. (This is the elliptic curve analog of computing the subgroup of roots of unity in a number field, see Algorithms 4.9.9 and 4.9.10.)

By considering the formal group associated with the elliptic curve, one can prove (see [Sil]) that torsion points of composite order in any number field have integral coordinates in any Weierstraß model with integral coefficients. Moreover, there are bounds on the denominators of the coordinates of torsion points of order p^n where p is a prime. Over $\mathbb{Q}$, these bounds tell us that only the points of order 2 may have non-integral coordinates in a generalized Weierstraß model, and in that case the denominator of the x-coordinate is at most 4. Using the fact that if P is a torsion point, then $2P$ is also one, one obtains the following theorem, due to Nagell and Lutz (see [Sil]).

Theorem 7.5.4 (Nagell-Lutz). *If $P = (x, y)$ is a rational point of finite order $n > 2$ on the elliptic curve $y^2 = x^3 + Ax + B$, where A and B are integers, then x and y are integers and y^2 divides the discriminant $-(4A^3 + 27B^2)$.*

This result, together with Mazur's Theorem 7.1.11 gives us the following algorithm.

Algorithm 7.5.5 (Rational Torsion Points). Given integers $a_1, \ldots, a_6$, this algorithm lists the rational torsion points on the corresponding elliptic curve E.

1. [2-Division Points] Using Formulas (7.1), compute b_2, b_4, b_6, b_8 and Δ. Output the origin of the curve ($(0:1:0)$ in projective coordinates). Set $P \leftarrow 4X^3 + b_2X^2 + 2b_4X + b_6$. For each rational root α of P, output the point $(\alpha, -(a_1\alpha + a_3)/2)$.
2. [Initialize Loop] Set $n \leftarrow 4\prod_{p|\Delta} p^{\lfloor v_p(\Delta)/2 \rfloor}$, the largest integer whose square divides 16Δ. Form the list $\mathcal{L}$ of all positive divisors of n.
3. [Loop on $2y + a_1x + a_3$] If $\mathcal{L}$ is empty, terminate the algorithm. Otherwise, let d be the smallest element of $\mathcal{L}$, and remove d from $\mathcal{L}$. For each rational root α of $P - d^2$ execute step 4, then go to step 3.
4. [Check if torsion] Set $P_1 \leftarrow (\alpha, (d - a_1\alpha - a_3)/2)$. Compute the points $2P_1$, $3P_1$, $4P_1$, $5P_1$ and $6P_1$, and let $x_2, \ldots, x_6$ be their x-coordinates. If one of these points is the origin of the curve, or if one of the x_i is equal to the x-coordinate of a point found in step 1, or if $x_2 = x_3$ or $x_3 = x_4$ or $x_4 = x_5$, then output the two points P_1 and $P_2 \leftarrow (\alpha, -(d + a_1\alpha + a_3)/2)$.

Indeed, from Mazur's Theorem 7.1.11, it is clear that P_1 will be a torsion point if and only if kP_1 is a point of order dividing 2 for $k \leq 6$ or if $kP_1 =$

$-(k+1)P_1$ for $k \leq 4$, and since opposite points have equal x-coordinates in a Weierstraß model, we deduce the test for torsion used in step 4.

Note that to obtain the torsion *subgroup* from this algorithm is very easy: if the polynomial P of step 1 has three rational roots, the torsion subgroup is isomorphic to $(\mathbb{Z}/2\mathbb{Z}) \times (\mathbb{Z}/(N/2)\mathbb{Z})$ otherwise it is isomorphic to $\mathbb{Z}/N\mathbb{Z}$, where N is the total number of torsion points output by the algorithm.

The last algorithm that we will see in this section is an algorithm to compute the canonical height of a rational point.

The Weil height of a point $P = (\frac{a}{e^2}, \frac{b}{e^3})$ on an elliptic curve E is defined to be $h(P) = \ln|e|$. It is known that the limit

$$\hat{h}(P) = \lim_{n \to \infty} \frac{h(2^n P)}{2^{2n}}$$

exists and defines a positive definite quadratic form on $\mathbb{R} \otimes E(\mathbb{Q})$, known as the canonical height function on $E(\mathbb{Q})$. The existence of this limit means that when a rational point with large denominator is multiplied by some integer m for the group law on the curve, the number of digits of its denominator is multiplied by m^2.

The symmetric bilinear form $\langle P, Q \rangle = \hat{h}(P+Q) - \hat{h}(P) - \hat{h}(Q)$ is called the canonical height pairing and is used to compute the regulator in the Birch and Swinnerton-Dyer Conjecture 7.3.9. The canonical height has properties analogous to those of the logarithmic embedding for number fields (Theorem 4.9.7). More precisely, $\hat{h}(P) = 0$ if and only if P is a point of finite order. More generally if $P_1, \ldots, P_r$ are points on E, then $\det(\langle P_i, P_j \rangle) = 0$ if and only if there exists a linear combination of the points (for the group law of E) which is a point of finite order. Hence this determinant is called the (elliptic) *regulator* of the points P_i.

If $P_1, \ldots, P_r$ form a *basis* of the torsion-free part of $E(\mathbb{Q})$, the regulator $R(E/\mathbb{Q})$ which enters in the Birch and Swinnerton-Dyer conjecture is the elliptic regulator of the points P_i.

The height function $\hat{h}(P)$ has a very interesting structure (see [Sil]). We will only note here that it can be expressed as a sum of local functions, one for each prime number p and one for the "Archimedean prime" ∞. To compute the contribution of a prime p we use an algorithm due in this form to Silverman (see [Sil2]). We will always assume that the elliptic curve is given by a global minimal equation, obtained for example by Algorithm 7.5.3.

Algorithm 7.5.6 (Finite part of the height). Given $a_1, \ldots, a_6 \in \mathbb{Z}$ the coefficients of the global minimal equation of an elliptic curve E and the coordinates (x, y) of a rational point P on E, this algorithm computes the contribution of the finite primes to the canonical height $\hat{h}(P)$.

1. [Initialize] Using Formulas (7.1), compute b_2, b_4, b_6, b_8, c_4, and Δ. Set $z \leftarrow (1/2)\ln(\text{denominator of } x)$, $A \leftarrow$numerator of $3x^2 + 2a_2x + a_4 - a_1y$,

$B \leftarrow$ numerator of $2y + a_1x + a_3$, $C \leftarrow$numerator of $3x^4 + b_2x^3 + 3b_4x^2 + 3b_6x + b_8$ and $D \leftarrow \gcd(A, B)$.

2. [Loop on p] If $D = 1$, output z and terminate the algorithm. Otherwise, choose a prime divisor p of D and set $D \leftarrow D/p$ until $p \nmid D$.

3. [Add local contribution] If $p \nmid c_4$, then set $N \leftarrow v_p(\Delta)$, $n \leftarrow \min(v_p(B), N/2)$ and $z \leftarrow z - (n(N-n)/(2N)) \ln p$. Otherwise, if $v_p(C) \geq 3v_p(B)$ set $z \leftarrow z - (v_p(B)/3) \ln p$ else set $z \leftarrow z - (v_p(C)/8) \ln p$. Go to step 2.

The Archimedean contribution has a more interesting history from the computational point of view. Initially, it was defined using logarithms of σ functions on the curve, but such objects are not easy to compute by hand or with a hand-held calculator. Tate then discovered a very nice way to compute it using a simple series. Silverman's paper [Sil2] also contains an improvement to that method. However, that series converges only geometrically (the $n-$th term is bounded by a constant times 4^{-n}). The original definition, while more cumbersome, has a faster rate of convergence by using q-expansions, so it should be preferred for high-precision calculations.

Algorithm 7.5.7 (Height Contribution at ∞). Given $a_1, \ldots, a_6 \in \mathbb{R}$ and the coordinates (x, y) of a point P on $E(\mathbb{R})$, this algorithm computes the Archimedean contribution of the canonical height of P.

1. [Initialize] Using Formulas (7.1), compute b_2, b_4, b_6 and Δ. Using Algorithm 7.4.7, compute ω_1 and ω_2. Using Algorithm 7.4.8, compute the elliptic logarithm z of the point P. Set $\lambda \leftarrow 2\pi/\omega_2$, $t \leftarrow \lambda \operatorname{Re}(z)$ and $q \leftarrow e^{2i\pi\omega_1/\omega_2}$. (Note that q is a real number and $|q| < 1$.)

2. [Compute theta function] Set

$$\theta \leftarrow \sum_{n=0}^{\infty} \sin((2n+1)t)(-1)^n q^{n(n+1)/2}$$

(stopping the sum when $q^{n(n+1)/2}$ becomes sufficiently small).

3. [Terminate] Output

$$\frac{1}{32} \ln \left| \frac{\Delta}{q} \right| + \frac{1}{8} \ln \left(\frac{x^3 + (b_2/4)x^2 + (b_4/2)x + b_6/4}{\lambda} \right) - \frac{1}{4} \ln |\theta|$$

and terminate the algorithm.

The canonical height $\hat{h}(P)$ is the sum of the two contributions coming from Algorithms 7.5.6 and 7.5.7.

7.5.3 Algorithms for computing the L-function

As we have seen, according to the Birch and Swinnerton-Dyer conjecture, most of the interesting arithmetical invariants of an elliptic curve E are grouped together in the behavior of $L(E,s)$ around the point $s=1$, in a manner similar to the case of number fields. In this section, we would like to explain how to compute this L function at $s=1$, assuming of course that E is a modular elliptic curve. The result is analogous to Propositions 5.3.14 and 5.6.11 but is in fact simpler since it (apparently) does not involve any higher transcendental functions.

Proposition 7.5.8. *Let E be a modular elliptic curve, let N be the conductor of E, let $L(E,s)=\sum_{n\geq 1}a_n n^{-s}$ be the L-series of E and finally let $\varepsilon=\pm 1$ be the sign in the functional equation for $L(E,s)$. Then if A is any positive real number, we have*

$$L(E,1)=\sum_{n=1}^{\infty}\frac{a_n}{n}\left(e^{-2\pi nA/\sqrt{N}}+\varepsilon e^{-2\pi n/(A\sqrt{N})}\right)$$

and in particular

$$L(E,1)=(1+\varepsilon)\sum_{n=1}^{\infty}\frac{a_n}{n}e^{-2\pi n/\sqrt{N}}.$$

As in the case of quadratic fields, we have given the general formula involving a real parameter A, but here the purpose is different. In the case of quadratic fields, it gave the possibility of checking the correctness of the computation of certain higher transcendental functions. Here, its use is very different: since the expression must be independent of A, it gives an indirect but quite efficient way to compute the sign ε (and also the conductor N for that matter), which otherwise is not so easy to compute (although there exist algorithms for doing so which are rather tedious). Indeed, we compute the right hand side of the formula giving $L(E,1)$ for two different values of A, say $A=1$ and $A=1.1$ (A should be close to 1 for optimal speed), and the results must agree. Only one of the two possible choices for ε will give results which agree. Hence the above proposition enables us, not only to compute $L(E,1)$ to great accuracy (the series converges exponentially) but also to determine the sign of the functional equation. Also note that the a_p are computed using Algorithm 7.4.12 or simply as a sum of Legendre symbols, and the a_n are computed using the relations $a_1=1$, $a_{mn}=a_m a_n$ if m and n are coprime, and $a_{p^k}=a_p a_{p^{k-1}}-pa_{p^{k-2}}$ for $k\geq 2$.

This is not the whole story. Assume that we discover in this way that $\varepsilon=-1$. Then $L(E,1)=0$ for trivial antisymmetry reasons, but the Birch and Swinnerton-Dyer conjecture tells us that the interesting quantity to compute

is now the derivative $L'(E,1)$ of $L(E,s)$ at $s=1$. In that case we have the following proposition which now involves higher transcendental functions.

Proposition 7.5.9. *Let E be a modular elliptic curve, let N be the conductor of E, and let $L(E,s)=\sum_{n\geq 1}a_n n^{-s}$ be the L-series of E. Assume that the sign ε of the functional equation for $L(E,s)$ is equal to -1 (hence trivially $L(E,1)=0$). Then*

$$L'(E,1)=2\sum_{n=1}^{\infty}\frac{a_n}{n}E_1\left(\frac{2\pi n}{\sqrt{N}}\right)$$

where E_1 is the exponential integral function already used in Proposition 5.6.11.

In the case where $L(E,s)$ vanishes to order greater than 1 around $s=1$, there exist similar formulas for $L^{(r)}(E,1)$ using functions generalizing the function $E_1(x)$. We refer to [BGZ] for details. If we assume the Birch and Swinnerton-Dyer conjecture, these formulas allow us to compute the rank of the curve E as the exact order of vanishing of $L(E,s)$ around $s=1$. Note that although the convergence of the series which are obtained is exponential, we need at least $O(\sqrt{N})$ terms before the partial sums start to become significantly close to the result, hence the limit of this method, as in the case of quadratic fields, is for N around 10^{10}. In particular, if we want to estimate the rank of elliptic curves having a much larger conductor, other methods must be used (still dependent on all standard conjectures). We refer to [Mes2] for details.

7.6 Algorithms for Elliptic Curves with Complex Multiplication

7.6.1 Computing the Complex Values of $j(\tau)$

We first describe an efficient way to compute the numerical value of the function $j(\tau)$ for $\tau\in\mathcal{H}$.

Note first that, as in most algorithms of this sort, it is worthwhile to have τ with the largest possible imaginary part, hence to use $j(\tau)=j(\gamma(\tau))$ for any $\gamma\in\mathrm{SL}_2(\mathbb{Z})$. For this, we use Algorithm 7.4.2.

After this preliminary step, there are numerous formulas available to us for computing $j(\tau)$, as is the case for all modular forms or functions. We could for example use Algorithm 7.4.3 for computing g_2 and g_3. It would also be possible to use formulas based on the use of the arithmetic-geometric mean which are quadratically convergent. This would be especially useful for high precision computations of $j(\tau)$.

We will use an intermediate approach which I believe is best suited for practical needs. It is based on the following formulas.

Set as usual $q = e^{2i\pi\tau}$, and

$$\Delta(\tau) = q\left(1 + \sum_{n\geq 1}(-1)^n\left(q^{n(3n-1)/2} + q^{n(3n+1)/2}\right)\right)^{24}.$$

This expression should be computed as written. Note that the convergence is considerably better than that of an ordinary power series since the exponents grow quadratically. It is a well known theorem on modular forms that

$$g_2^3 - 27g_3^2 = \left(\frac{2\pi}{\omega_2}\right)^{12}\Delta.$$

Now the formula that we will use for computing $j(\tau)$ is

$$j(\tau) = \frac{(256f(\tau)+1)^3}{f(\tau)} \quad \text{where} \quad f(\tau) = \frac{\Delta(2\tau)}{\Delta(\tau)}$$

(note that changing τ into 2τ changes q into q^2).

7.6.2 Computing the Hilbert Class Polynomials

Our second goal is to compute the equation of degree $h(D)$ satisfied by $j(\tau)$, which we will call the *Hilbert class polynomial* for the discriminant D. For this we directly apply Theorem 7.2.14. This leads to the following algorithm, which is closely modeled on Algorithm 5.3.5.

Algorithm 7.6.1 (Hilbert Class Polynomial). Given a negative discriminant D, this algorithm computes the monic polynomial of degree $h(D)$ in $\mathbb{Z}[X]$ of which $j((D+\sqrt{D})/2)$ is a root. We make use of a polynomial variable P.

1. [Initialize] Set $P \leftarrow 1$, $b \leftarrow D \bmod 2$ and $B \leftarrow \left\lfloor\sqrt{|D|/3}\right\rfloor$.
2. [Initialize a] Set $t \leftarrow (b^2-D)/4$ and $a \leftarrow \max(b,1)$.
3. [Test] If $a \nmid t$ go to step 4. Otherwise compute $j \leftarrow j((-b+\sqrt{D})/(2a))$ using the above formulas. Now if $a = b$ or $a^2 = t$ or $b = 0$ set $P \leftarrow P\cdot(X-j)$, else set $P \leftarrow P\cdot(X^2 - 2\,\mathrm{Re}(j)X + |j|^2)$.
4. [Loop on a] Set $a \leftarrow a+1$. If $a^2 \leq t$, go to step 3.
5. [Loop on b] Set $b \leftarrow b+2$. If $b \leq B$ go to step 2, otherwise round the coefficients of P to the nearest integer, output P and terminate the algorithm.

An important remark must be made, otherwise this algorithm would not make much sense. The final coefficients of P (known to be integers) must be

computed within an error of 0.5 at most. For this, we need to make some a priori estimate on the size of the coefficients of P. In practice, we look at the constant term, which is usually not far from being the largest. This term is equal to the product of the values $j((-b+\sqrt{D})/(2a))$ over all reduced forms (a,b,c), and the modulus of this is approximately equal to $e^{\pi\sqrt{|D|}/(2a)}$ hence the modulus of the constant term is relatively close to 10^k, where

$$k = \frac{\pi\sqrt{|D|}}{\ln(10)} \sum \frac{1}{a},$$

the sum running over all reduced forms (a,b,c) of discriminant D.

Hence in step 3, the computation of the j-values should be done with at least $k+10$ significant digits, 10 being an empirical constant which is sufficient in practice. Note that the value of $\sum 1/a$ is not known in advance, so it should be computed independently (by again applying a variant of Algorithm 5.3.5), since this will in any case take a negligible proportion of the time spent.

7.6.3 Computing Weber Class Polynomials

One of the main applications of computing the Hilbert class polynomials is to explicitly generate the *Hilbert class field* of $K = \mathbb{Q}(\sqrt{D})$ when D is a negative fundamental discriminant. As already mentioned, the coefficients of these polynomials will be very large, and it is desirable to make them smaller. One method is to use the POLRED Algorithm 4.4.11. An essentially equivalent method is given in [Kal-Yui]. A better method is to start by using some extra algebraic information.

We give an example. Set

$$\eta(\tau) = e^{2i\pi\tau/24}\left(1 + \sum_{n\geq 1}(-1)^n\left(q^{n(3n-1)/2} + q^{n(3n+1)/2}\right)\right)$$

(this is the 24-th root of the function $\Delta(\tau)$ defined above, and is called *Dedekind's eta-function*). Define

$$f_1(\tau) = \frac{\eta(\tau/2)}{\eta(\tau)}.$$

Then if $D \equiv \pm 8 \pmod{32}$ and $3 \nmid D$, if we set

$$u = f_1(\sqrt{D/4})^2\sqrt{2},$$

we can use u instead of j for generating the class field. Indeed, one can show that $K(j) = K(u)$, that u is an algebraic integer (of degree equal to $h(D)$), and what is more important, that the coefficients of the minimal monic polynomial

of u (which we will call the *Weber class polynomial* for D) have approximately 12 times fewer digits than those of the Hilbert class polynomials.

Note that one can easily recover j from u if needed. For example, in our special case above we have

$$j = \frac{(256 - u^{12})^3}{u^{24}}.$$

This takes care only of certain congruence classes for D, but most can be treated in a similar manner. We refer the interested reader to [Atk-Mor] or to [Kal-Yui] for complete details.

The algorithm for computing the Weber class polynomials is essentially identical to the one for Hilbert class polynomials: we replace j by u, and furthermore use a much lower precision for the computation of u. For example, in the case $D \equiv \pm 8 \pmod{32}$ and $3 \nmid D$, we can take approximately one twelfth of the number of digits that were needed for the Hilbert class polynomials.

7.7 Exercises for Chapter 7

1. (J. Cremona) Given c_4 and c_6 computed by Formulas (7.1), we would like to recover the b_i and a_i, where we assume that the a_i are in $\mathbb{Z}$. Show that the following procedure is valid. Let b_2 be the unique integer such that $-5 \leq b_2 \leq 6$ and $b_2 \equiv -c_6 \bmod 12$. Then set $b_4 = (b_2^2 - c_4)/24$, $b_6 = (-b_2^3 + 36b_2b_4 - c_6)/216$. Finally set $a_1 = b_2 \bmod 2 \in \{0,1\}$, $a_2 = (b_2 - a_1)/4 \in \{-1,0,1\}$, $a_3 = b_6 \bmod 2 \in \{0,1\}$, $a_4 = (b_4 - a_1a_3)/2$ and $a_6 = (b_6 - a_3)/4$.

2. Let E be an elliptic curve with complex multiplication by the complex quadratic order of discriminant D. Show that if p is a prime such that $\left(\frac{D}{p}\right) = -1$, then $|E(\mathbb{Z}/p\mathbb{Z})| = p + 1$.

3. Using the result of Exercise 2, show that the only torsion points on the elliptic curve $y^2 = x^3 - n^2x$ (which has complex multiplication by $\mathbb{Z}[i]$) are the 4 points of order 1 or 2. (Hint: use Dirichlet's theorem on the infinitude of primes in arithmetic progressions.)

4. Show that the elliptic curve $y^2 = 4x^3 - 30x - 28$ has complex multiplication by $\mathbb{Z}[\sqrt{-2}]$ and give explicitly the action of multiplication by $\sqrt{-2}$ on a point (x, y).

5. Given an elliptic curve defined over $\mathbb{Q}$ by a generalized Weierstraß equation, write an algorithm which determines whether this curve has complex multiplication, and if this is the case, gives the complex quadratic order $\mathrm{End}(E)$. (This exercise requires some additional knowledge about elliptic curves.)

6. Using Algorithm 7.4.10, find a Weierstraß equation for the elliptic curve E given by the projective equation

$$x^3 + y^3 = dt^3$$

with $(1 : -1 : 0)$ as given rational point.

7. Given the point $(2 : 1 : 1)$ on the elliptic curve whose projective equation is $x^3+y^3=9t^3$, find another rational point with positive coordinates (apart from the point $(1 : 2 : 1)$ of course). It may be useful to use the result of Exercise 6.

8. Given an elliptic curve E by a general Weierstraß equation $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$ and a complex number z, give the formulas generalizing those of Proposition 7.4.4 for the coordinates (x,y) on $E(\mathbb{C})$ corresponding to z considered as an element of $\mathbb{C}/L$ where L is the lattice associated to E.

9. (J.-F. Mestre) Let r_1, r_2, r_3 and r_4 be distinct rational numbers and let t be a parameter (which we will also take to be a rational number). Consider the polynomial of degree 12

$$P(X)=\prod_{1\le i,j\le 4,\ i\ne j}(X-(r_i+tr_j)).$$

a) By considering the Laurent series expansion of $P^{1/3}$ show that for any monic polynomial P of degree 12 there exists a unique polynomial $g\in\mathbb{Q}[X]$ such that $\deg(P(X)-g^3(X))\le 7$, and show that in our special case we have in fact $\deg(P(X)-g^3(X))\le 6$.

b) Show that there exists $q(X)\in\mathbb{Q}[X]$ and $r(X)\in\mathbb{Q}[X]$ such that $P(X)=g^3(X)+q(X)g(X)+r(X)$ with $\deg(q)\le 2$ and $\deg(r)\le 3$.

c) Deduce from this that the equation $Y^3+q(X)Y+r(X)=0$ is the equation of a cubic with rational coefficients, and that the 12 points $(r_i+tr_j, g(r_i+tr_j))_{i\ne j}$ are 12 (not necessarily distinct) rational points on this cubic.

d) Give explicit values of the r_i and t such that the cubic is non-singular, the 12 points above are distinct and in fact linearly independent for the group law on the cubic.

e) Using Algorithm 7.4.10, find a Weierstraß equation corresponding to the cubic, and give explicitly an elliptic curve defined over $\mathbb{Q}$ whose rank is at least equal to 11 as well as 11 independent points on the elliptic curve (note that we have to "lose" a point in order to obtain an elliptic curve). To answer the last two questions of this exercise, the reader is strongly advised to use a package such as those described in Appendix A. In [Nag] it is shown how to refine this construction in order to have infinite families of elliptic curves of rank 13 instead of 11.

10. Prove that the AGM of two positive real numbers exists, i.e. that the two sequences a_n and b_n given in the text both converge and to the same limit. Show also that the convergence is quadratic.

11. The goal of this exercise is to prove the formula giving $\mathrm{AGM}(a,b)$ in terms of an elliptic integral.

a) Set

$$I(a,b)=\int_0^{\pi/2}\frac{dt}{\sqrt{a^2\cos^2 t+b^2\sin^2 t}}.$$

By making the change of variable $\sin t=2a\sin u/((a+b)+(a-b)\sin^2 u)$ show that $I(a,b)=I((a+b)/2,\sqrt{ab})$.

b) Deduce from this the formula $I(a,b)=\pi/(2\,\mathrm{AGM}(a,b))$ given in the text.

c) By making the change of variable $x=a+(b-a)\sin^2 t$, express $I(a,b)$ as an elliptic integral.

Chapter 8

Factoring in the Dark Ages

I owe this title to a talk given by Hendrik Lenstra at MSRI Berkeley in the spring of 1990.

8.1 Factoring and Primality Testing

Since Fermat, it is known that the problem of decomposing a positive integer N into the product of its prime factors splits in fact in three subproblems. The first problem is to decide quickly whether N is composite or probably prime. Such tests, giving a correct answer when N is composite, but no real answer when N is prime, will be called *compositeness tests* (and certainly not primality tests). We will study them in Section 8.2. The second problem is, if one is almost sure that N is prime, to prove that it is indeed prime. Methods used before 1980 to do this will be studied in Section 8.3. Modern methods are the subject matter of Chapter 9. The third problem is that once one knows that N is composite, to factor N. Methods used before the 1960's (i.e. in the dark ages) will be studied starting at Section 8.4. Modern methods are the subject matter of Chapter 10.

Note that factoring/primality testing is usually a recursive process. Given a composite number N, a factoring method will not in general give the complete factorization of N, but only a non-trivial factor d, i.e. such that $1 < d < N$. One then starts working on the two pieces d and N/d. Finding a non-trivial divisor d of N will be called *splitting* N, or even, sometimes by abuse of language, factoring N.

Before going to the next section, it should be mentioned that the most naïve method of trial division (which simultaneously does factoring and primality testing) deserves a paragraph. Indeed, in most factoring methods, it usually never hurts to trial divide up to a certain bound to remove small factors. Now we want to divide N by primes up to the square root of N. For this, we may or may not have at our disposal a sufficiently large table of primes. If this is not the case, it is clear that we can divide N by numbers d in given congruence classes, for example 1 and 5 modulo 6, or 1, 7, 11, 13, 17, 19, 23, 29 modulo 30. We will then make unnecessary divisions (by composite numbers), but the result will still be correct. Hence we may for instance use the following algorithm.

Algorithm 8.1.1 (Trial Division). We assume given a table of prime numbers $p[1] = 2, p[2] = 3, \ldots, p[k]$, with $k > 3$, an array $t \leftarrow [6, 4, 2, 4, 2, 4, 6, 2]$, and an index j such that if $p[k] \bmod 30$ is equal to 1, 7, 11, 13, 17, 19, 23 or 29 then j is set equal to equal to 0, 1, 2, 3, 4, 5, 6 or 7 respectively. Finally, we give ourselves an upper bound B such that $B \geq p[k]$, essentially to avoid spending too much time.

Then given a positive integer N, this algorithm tries to factor (or split N), and if it fails, N will be free of prime factors less than or equal to B.

1. [Initialize] If $N \leq 5$, output the factorization $1 = 1$, $2 = 2$, $3 = 3$, $4 = 2^2$, $5 = 5$ corresponding to the value of N, and terminate the algorithm. Otherwise, set $i \leftarrow -1$, $m \leftarrow 0$, $l \leftarrow \lfloor \sqrt{N} \rfloor$.

2. [Next prime] Set $m \leftarrow m + 1$. If $m > k$ set $i \leftarrow j - 1$ and go to step 5, otherwise set $d \leftarrow p[m]$.

3. [Trial divide] Set $r \leftarrow N \bmod d$. If $r = 0$, then output d as a non-trivial divisor of N and terminate the algorithm (or set $N \leftarrow N/d$, $l \leftarrow \lfloor \sqrt{N} \rfloor$ and repeat step 3 if we want to continue finding factors of N).

4. [Prime?] If $d \geq l$, then if $N > 1$ output a message saying that the remaining N is prime and terminate the algorithm. Otherwise, if $i < 0$ go to step 2.

5. [Next divisor] Set $i \leftarrow i + 1 \bmod 8$, $d \leftarrow d + t[i]$. If $d > B$, then output a message saying that the remaining prime divisors of N are greater than B, otherwise go to step 3.

Note that we have $i = -1$ as long as we are using our prime number table, $i \geq 0$ if not.

This test should not be used for factoring completely, except when N is very small (say $N < 10^8$) since better methods are available for that purpose. On the other hand, it is definitely useful for removing small factors.

Implementation Remark. I suggest using a table of primes up to 500000, if you can spare the memory (this represents 41538 prime numbers). Trial division up to this limit usually never takes more than a few seconds on modern computers. Furthermore, only the difference of the primes (or even half of these differences) should be stored and not the primes themselves, since $p[k] - p[k-1]$ can be held in one byte instead of four when $p[k] \leq 436273009$, and $(p[k] - p[k-1])/2$ can be held in one byte if $p[k] \leq 304599508537$ (see [Bre3]).

Also, I suggest not doing any more divisions after exhausting the table of primes since there are better methods to remove small prime factors. Finally, note that it is not really necessary to compute $l \leftarrow \lfloor \sqrt{N} \rfloor$ in the initialization step, since the test $d \geq l$ in step 4 can be replaced by the test $q \leq l$, where q is the Euclidean quotient of N by d usually computed simultaneously with the remainder in step 3.

8.2 Compositeness Tests

The first thing to do after trial dividing a number N up to a certain bound, is to check whether N (or what remains of the unfactored part) is probably prime or composite. The possibility of doing this easily is due to Fermat's theorem $a^{p-1} \equiv 1 \pmod{p}$ when p is a prime not dividing a. Fermat's theorem in itself would not be sufficient however, even for getting a probable answer.

The second reason Fermat's theorem is useful is that $a^{p-1} \bmod p$ can be computed quickly using the powering algorithms of Section 1.2. This is in contrast with for instance Wilson's theorem stating that $(p-1)! \equiv -1 \pmod{p}$ if and only if p is prime. Although superficially more attractive than Fermat's theorem since it gives a necessary and sufficient condition for primality, and not only a necessary one, it is totally useless because nobody knows how to compute $(p-1)! \bmod p$ in a reasonable amount of time.

The third reason for the usefulness of Fermat's theorem is that although it gives only a necessary condition for primality, exceptions (i.e. composite numbers which satisfy the theorem) are rare. They exist, however. For example the number $N = 561 = 3 \cdot 11 \cdot 17$ is such that $a^{N-1} \equiv 1 \pmod{N}$ as soon as $(a, N) = 1$. Such numbers are called Carmichael numbers. It has just recently been proved by Alford, Granville and Pomerance ([AGP]) that there are infinitely many Carmichael numbers and even that up to x their number is at least $C \cdot x^{2/7}$ for some positive constant C.

It is not difficult to strengthen Fermat's theorem. If p is an odd prime and p does not divide a, then $a^{(p-1)/2} \equiv \pm 1 \pmod{p}$ (more precisely it is congruent to the Legendre symbol $\left(\frac{a}{p}\right)$, see Section 1.4.2). This is stronger than Fermat, and for example eliminates 561. It does not however eliminate all counter-examples, since for instance $N = 1729$ satisfies $a^{(N-1)/2} \equiv 1 \pmod{N}$ for all a coprime to N.

The first test which is really useful is due to Solovay and Strassen ([Sol-Str]). It is based on the fact that if we require not only $a^{(N-1)/2} \equiv \pm 1 \pmod{N}$ but $a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \pmod{N}$, where $\left(\frac{a}{N}\right)$ is the Jacobi-Kronecker symbol, then this will be satisfied by at most $N/2$ values of a when N is not a prime. This gives rise to the first compositeness test, which is probabilistic in nature: for 50 (say) randomly chosen values of a, test whether the congruence is satisfied. If it is not for any value of a, then N is composite. If it is for all 50 values, then we say that N is probably prime, with probability of error less than $2^{-50} \approx 10^{-15}$, lower in general than the probability of a hardware error.

This test has been superseded by a test due to Miller and Rabin ([Mil], [Rab]), which has two advantages. First, it does not require any Jacobi symbol computation, and second the number of a which will satisfy the test will be at most $N/4$ instead of $N/2$, hence fewer trials have to be made to ensure a given probability. In addition, one can prove that if a satisfies the Rabin-Miller test, then it will also satisfy the Solovay-Strassen test, so the Miller-Rabin test completely supersedes the Solovay-Strassen test.

Definition 8.2.1. *Let N be an odd positive integer, and a be an integer. Write $N-1=2^t q$ with q odd. We say that N is a* strong pseudo-prime *in base a if either $a^q \equiv 1 \pmod{N}$, or if there exists an e such that $0 \leq e < t$ and $a^{2^e q} \equiv -1 \pmod{N}$.*

If p is an odd prime, it is easy to see that p is a strong pseudo-prime in any base not divisible by p (see Exercise 1). Conversely, one can prove (see for example [Knu2]) that if p is not prime, there exist less than $p/4$ bases a such that $1 < a < p$ for which p is a strong pseudo-prime in base a. This leads to the following algorithm.

Algorithm 8.2.2 (Rabin-Miller). Given an odd integer $N \geq 3$, this algorithm determines with high probability if N is composite. If it fails, it will output a message saying that N is probably prime.

1. [Initialize] Set $q \leftarrow N-1$, $t \leftarrow 0$, and while q is even set $q \leftarrow q/2$ and $t \leftarrow t+1$ (now $N-1=2^t q$ with q odd). Then set $c \leftarrow 20$.
2. [Choose new a] Using a random number generator, choose randomly an a such that $1 < a < N$. Then set $e \leftarrow 0$, $b \leftarrow a^q \bmod N$. If $b = 1$, go to step 4.
3. [Squarings] While $b \not\equiv \pm 1 \pmod{N}$ and $e \leq t-2$ set $b \leftarrow b^2 \bmod N$ and $e \leftarrow e+1$. If $b \not\equiv N-1$ output a message saying that N is composite and terminate the algorithm.
4. [Repeat test] Set $c \leftarrow c-1$. If $c > 0$ go to step 2, otherwise output a message saying that N is probably prime.

The running time of this algorithm is essentially the same as that of the powering algorithm which is used, i.e. in principle $O(\ln^3 N)$. Note however that we can reasonably restrict ourselves to single precision values of a (which will not be random any more, but it probably does not matter), and in that case if we use the left-right Algorithms (1.2.2 to 1.2.4), the time drops to $O(\ln^2 N)$. Hence, it is essentially as fast as one could hope for.

This algorithm is the workhorse of compositeness tests, and belongs in almost any number theory program. Note once again that it will prove the compositeness of essentially all numbers, but it will never prove their primality. In fact, by purely theoretical means, it is usually possible to construct composite numbers which pass the Rabin-Miller test for any given reasonably small finite set of bases a ([Arn]). For example, the composite number

$$\begin{aligned} &1195068768795265792518361315725116351898245581 \\ &\qquad = 2444451644843139244746 1 \cdot 48889032896862784894921 \end{aligned}$$

is a strong pseudo-prime to bases 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 and 31 and several others.

There is a variation on this test due to Miller which is as follows. If one assumes the Generalized Riemann Hypothesis, then one can prove that if N

is not prime, there exists an $a < C\ln^2 N$ such that N will not be a strong pseudo-prime in base a, C being an explicit constant. Hence this gives a non-probabilistic primality and compositeness test, but since it is based on an unproven hypothesis, it cannot be used for the moment. Note that the situation is completely different in factoring algorithms. There, we can use any kinds of unproven hypotheses or crystal balls for that matter, since once the algorithm (or pseudo-algorithm) finishes, one can immediately check whether we have indeed obtained a factor of our number N, without worrying about the manner in which it was obtained. Primality testing however requires rigorous mathematical proofs.

Note also that even if one uses the best known values of the constant C, for our typical range of values of N (say up to 10^{500}), the modern methods explained in Chapter 9 are in practice faster.

8.3 Primality Tests

We now consider the practical problem of rigorously *proving* that a number N is prime. Of course, we will try to do this only after N has successfully passed the Rabin-Miller test, so that we are morally certain that N is indeed prime.

8.3.1 The Pocklington-Lehmer $N-1$ Test

We need a sort of converse to Fermat's theorem. One such converse was found by Pocklington, and improved by Lehmer. It is based on the following result.

Proposition 8.3.1. *Let N be a positive integer, and let p be a prime divisor of $N-1$. Assume that we can find an integer a_p such that $a_p^{N-1} \equiv 1 \pmod{N}$ and $(a_p^{(N-1)/p} - 1, N) = 1$. Then if d is any divisor of N, we have $d \equiv 1 \pmod{p^{\alpha_p}}$, where p^{α_p} is the largest power of p which divides $N-1$.*

Proof. It is clearly enough to prove the result for all prime divisors of N, since any divisor is a product of prime divisors. Now if d is a prime divisor of N, we have $a_p^{d-1} \equiv 1 \pmod{d}$, since a_p is coprime to N (why?) hence to d. On the other hand, since $(a_p^{(N-1)/p} - 1, N) = 1$, we have $a_p^{(N-1)/p} \not\equiv 1 \pmod{d}$. If e is the exact order of a_p modulo d (i.e. the smallest positive exponent such that $a_p^e \equiv 1 \pmod{d}$), this means that $e \mid d-1$, $e \nmid (N-1)/p$ but $e \mid N-1$, hence $p^{\alpha_p} \mid e \mid d-1$ showing that $d \equiv 1 \pmod{p^{\alpha_p}}$. □

Corollary 8.3.2. *Assume that we can write $N-1 = F \cdot U$ where $(F, U) = 1$, F is completely factored, and $F > \sqrt{N}$. Then, if for each prime p dividing F we can find an a_p satisfying the conditions of Proposition 8.3.1, N is prime. Conversely, if N is prime, for any prime p dividing $N-1$, one can find a_p satisfying the conditions of Proposition 8.3.1.*

Proof. If the hypotheses of this corollary are satisfied, it follows immediately from Proposition 8.3.1 that all divisors of N are congruent to 1 mod F. Since $F > \sqrt{N}$, this means that N has no prime divisor less than its square root, hence N is prime.

Conversely, when N is prime, if we take for a_p a *primitive root* modulo N, i.e. a generator of the multiplicative group $(\mathbb{Z}/N\mathbb{Z})^*$, it is clear that the conditions of the proposition are satisfied since the order of a_p is exactly equal to $N-1$. □

This corollary gives us our first true primality test. Its main drawback is that we need to be able to factor $N-1$ sufficiently, and this is in general very difficult. It is however quite useful for numbers having special forms where $N-1$ factors easily, for example the Fermat numbers $2^{2^k}+1$ (see Exercise 9).

The condition $F > \sqrt{N}$ of the corollary can be weakened if we make an extra test:

Proposition 8.3.3. *Assume that we can write $N-1 = F \cdot U$ where $(F,U) = 1$, F is completely factored, all the prime divisors of U are greater than B, and $B \cdot F \geq \sqrt{N}$. Then if for each prime p dividing F we can find an a_p satisfying the conditions of Proposition 8.3.1, and if in addition we can find a_U such that $a_U^{N-1} \equiv 1 \pmod{N}$ and $(a_U^F - 1, N) = 1$, then N is prime. Conversely, if N is prime, such a_p and a_U can always be found.*

Proof. We follow closely the proof of Proposition 8.3.1. Let d be any prime divisor of N. Proposition 8.3.1 tells us that $d \equiv 1 \pmod{F}$. If e is the exact order of a_U modulo d, then $e \mid d-1$, $e \mid N-1$ and $e \nmid F = (N-1)/U$. Now one cannot have $(e,U) = 1$, otherwise from $e \mid N-1 = FU$ one would get $e \mid F$, contrary to the hypothesis. Hence $(e,U) > 1$, and since U has all its prime factors greater than B, $(e,U) > B$. Finally, since $(F,U) = 1$, from $d \equiv 1 \pmod{e}$ and $d \equiv 1 \pmod{F}$ we obtain $d \equiv 1 \pmod{(e,U) \cdot F}$ hence $d > B \cdot F \geq \sqrt{N}$, showing that N has no prime divisor less than or equal to its square root, hence that N is prime. □

Note that the condition that U has all its prime factors greater than B is very natural in practice since the factorization $N-1 = F \cdot U$ is often obtained by trial division.

8.3.2 Briefly, Other Tests

Several important generalizations of this test exist. First, working in the multiplicative group of the field $\mathbb{F}_{N^2}$ instead of $\mathbb{F}_N$, one obtains a test which uses the factorization of $N+1$ instead of $N-1$. This gives as a special case the Lucas-Lehmer test for Mersenne numbers $N = 2^p - 1$. In addition, since $\mathbb{F}_N$ is a subfield of $\mathbb{F}_{N^2}$, it is reasonable to expect that one can combine the information coming from the two tests, and this is indeed the case. One can

also use higher degree finite fields ($\mathbb{F}_{N^3}$, $\mathbb{F}_{N^4}$ and $\mathbb{F}_{N^6}$) which correspond to using in addition the completely factored part of N^2+N+1, N^2+1 and N^2-N+1 respectively. These numbers are already much larger, however, and do not always give much extra information. Other finite fields give even larger numbers. One last improvement is that, as in Proposition 8.3.3 one can use the upper bound used in doing the trial divisions to find the factors of $N-1$, $N+1$, etc ... For details, I refer to [BLS], [Sel-Wun] or [Wil-Jud].

8.4 Lehman's Method

We now turn our attention to factoring methods. The spirit here will be quite different. For example, we do not need to be completely rigorous since if we find a number which may be a factor of N, it will always be trivial to check if it is or not. It will however be useful to have some understanding of the asymptotic behavior of the algorithm.

Although several methods were introduced to improve trial division (which is, we recall, a $O(N^{1/2+\epsilon})$ algorithm), the first method which has a running time which could be proved to be substantially lower was introduced by Lehman (see [Leh1]). Its execution time is at worst $O(N^{1/3+\epsilon})$, and it is indeed faster than trial division already for reasonably small values of N. The algorithm is as follows.

Algorithm 8.4.1 (Lehman). Given an integer $N \geq 3$, this algorithm finds a non-trivial factor of N if N is not prime, or shows that N is prime.

1. [Trial division] Set $B \leftarrow \lfloor N^{1/3} \rfloor$. Trial divide N up to the bound B using Algorithm 8.1.1. If any non-trivial factor is found, output it and terminate the algorithm. Otherwise set $k \leftarrow 0$.
2. [Loop on k] Set $k \leftarrow k+1$. If $k > B$, output the fact that N is prime and terminate the algorithm. Otherwise, set $r = 1$ and $m = 2$ if k is even, $r = k+N$ and $m = 4$ if k is odd.
3. [Loop on a] For all integers a such that $4kN \leq a^2 \leq 4kN + B^2$ and $a \equiv r \pmod{m}$ do as follows. Set $c \leftarrow a^2 - 4kN$. Using Algorithm 1.7.3, test whether c is a square. If it is, let $c = b^2$, output $\gcd(a+b, N)$ (which will be a non-trivial divisor of N) and terminate the algorithm. Otherwise, use the next value of a if any. If all possible values of a have been tested, go to step 2.

Proof (D. Zagier). We only give a sketch, leaving the details as an exercise to the reader.

If no factors are found during step 1, this means that all the prime factors of N are greater than $N^{1/3}$ hence N has at most two prime factors.

Assume first that N is prime. Then the test in step 3 can never succeed. Indeed, if $a^2 - 4kN = b^2$ then $N \mid a^2 - b^2$ hence $N \mid (a-b)$ or $N \mid (a+b)$ so $a+b \geq N$, but this is impossible since the given inequalities on k and a imply

that $a < 2N^{2/3} + 1$ and $b < N^{1/3}$ so $N \leq 13$. An easy check shows that for $3 \leq N \leq 13$, N prime, the test in step 3 does not succeed.

Assume now that N is composite, so that $N = pq$ with p and q not necessarily distinct primes, where we may assume that $p \leq q$. Consider the convergents u_n/v_n of the continued fraction expansion of q/p. Let n be the unique index such that $u_n v_n < N^{1/3} < u_{n+1}v_{n+1}$ (which exists since $pq > N^{1/3}$). Using the elementary properties of continued fractions, if we set $k = u_n v_n$ and $a = pv_n + qu_n$, it is easily checked that the conditions of step 3 are met, thus proving the validity of the algorithm. □

For each value of k there are at most $1/2(\sqrt{4kN + N^{2/3}} - \sqrt{4kN}) \approx N^{1/6}k^{-1/2}/8$ values of a, and since $\sum_{k \leq x} k^{-1/2} \approx 2x^{1/2}$, the running time of the algorithm is indeed $O(N^{1/3+\epsilon})$ as claimed.

We refer to [Leh1] for ways of fine tuning this algorithm, which is now only of historical interest.

8.5 Pollard's ρ Method

8.5.1 Outline of the Method

The idea behind this method is the following. Let $f(X)$ be a polynomial with integer coefficients. We define a sequence by taking any initial x_0, and setting $x_{k+1} = f(x_k) \bmod N$. If p is a (unknown) prime divisor of N, then the sequence $y_k = x_k \bmod p$ satisfies the same recursion. Now if $f(X)$ is chosen suitably, it is not unreasonable to assume that this sequence will behave like the sequence of iterates of a *random* map from $\mathbb{Z}/p\mathbb{Z}$ into itself. Such a sequence must of course be ultimately periodic, and a mathematical analysis shows that it is reasonable to expect that the period and preperiod will have length $O(\sqrt{p})$. Now if $y_{k+t} = y_k$, this means that $x_{k+t} \equiv x_k \pmod{p}$, hence that $(x_{k+t} - x_k, N) > 1$. Now this GCD will rarely be equal to N itself, hence we obtain in this way, maybe not p, but a non-trivial factor of N, so N is split and we can look at the pieces. The number of necessary steps will be $O(\sqrt{p}) = O(N^{1/4})$, and the total time in bit operations will be $O(N^{1/4} \ln^2 N)$.

Of course, we have just given a rough outline of the method. It is clear however that it will be efficient since the basic operations are simple, and furthermore that its running time depends mostly on the size of the smallest prime factor of N, not on the size of N itself, hence it can replace trial division or Lehman's method to cast out small factors. In fact, it is still used along with more powerful methods for that purpose. Finally, notice that, at least in a primitive form, it is very easy to implement.

We must now solve a few related problems:

(1) How does one find the periodicity relation $y_{k+t} = y_k$?
(2) How does one choose f and x_0?
(3) What is the expected average running time, assuming f is a random map?

I would like to point out immediately that although it is believed that the polynomials that we give below behave like random maps, this is not at all proved, and in fact the exact mathematical statement to prove needs to be made more precise.

8.5.2 Methods for Detecting Periodicity

From now on, we consider a sequence $y_{k+1} = f(y_k)$ from a finite set E into itself. Such a sequence will be ultimately periodic, i.e. there exists M and $T > 0$ such that for $k \geq M$, $y_{k+T} = y_k$ but $y_{M-1+T} \neq y_{M-1}$. The number M will be called the preperiod, and T (chosen as small as possible) will be the period. If the iterates are drawn on a piece of paper starting at the bottom and ending in a circle the figure that one obtains has the shape of the Greek letter ρ, whence the name of the method.

We would like to find a reasonably efficient method for finding k and $t > 0$ such that $y_{k+t} = y_k$ (we do not need to compute M and T). The initial method suggested by Pollard and Floyd is to compute simultaneously with the sequence y_k the sequence z_k defined by $z_0 = y_0$, $z_{k+1} = f(f(z_k))$. Clearly $z_k = y_{2k}$, and if k is any multiple of T which is larger than M, we must have $z_k = y_{2k} = y_k$, hence our problem is solved. This leads to a simple-minded but nonetheless efficient version of Pollard's ρ method. Unfortunately we need three function evaluations per step, and this may seem too many.

An improvement due to Brent is the following. Let $l(m)$ be the largest power of 2 less than or equal to m, i.e.

$$l(m) = 2^{\lfloor \lg m \rfloor},$$

so that in particular $l(m) \leq m < 2l(m)$. Then I claim that there exists an m such that $y_m = y_{l(m)-1}$. Indeed, if one chooses

$$m = 2^{\lceil \lg \max(M+1,T) \rceil} + T - 1,$$

we clearly have $l(m) = 2^{\lceil \lg \max(M+1,T) \rceil}$ hence $l(m) - 1 \geq M$ and $m - (l(m) - 1) = T$, thus proving our claim.

If instead of computing an extra sequence z_k we compute only the sequence y_k and keep y_{2^e-1} each time we hit a power of two minus one, for every m such that $2^e \leq m < 2^{e+1}$ it will be enough to compare y_m with y_{2^e-1} (note that at any time there is only one value of y to be kept).

Hence Brent's method at first seems definitely superior. It can however be shown that the number of comparisons needed before finding an equality $y_m = y_{l(m)-1}$ will be on average almost double that of the initial Pollard-Floyd method. In practice this means that the methods are comparable, the lower number of function evaluations being compensated by the increased number of comparisons which are needed.

However a modification of Brent's method gives results which are generally better than the above two methods. It is based on the following proposition.

Proposition 8.5.1.

(1) *There exists an m such that*

$$y_m = y_{l(m)-1} \qquad \text{and} \qquad \frac{3}{2}l(m) \leq m < 2l(m).$$

(2) *the least such m is $m_0 = 3$ if $M = 0$ and $T = 1$ (i.e. if $y_1 = y_0$), and otherwise is given by*

$$m_0 = 2^{\lceil \lg \max(M+1,T) \rceil} + T\left\lceil \frac{l(M)+1}{T} \right\rceil - 1,$$

where we set $l(0) = 0$.

Proof. Set $e = \lceil \lg \max(M+1, T) \rceil$. We claim that, as in Brent's original method, we still have $l(m_0) = 2^e$. Clearly, $2^e \leq m_0$, so we must prove that $m_0 < 2^{e+1}$ or equivalently that

$$T\left\lceil \frac{l(M)+1}{T} \right\rceil \leq 2^e.$$

We consider two cases. First, if $T \leq l(M)$, then

$$T\left\lceil \frac{l(M)+1}{T} \right\rceil \leq l(M) + T \leq 2l(M) = 2^{\lceil \lg(M+1) \rceil} \leq 2^e,$$

since $\lfloor \lg M \rfloor + 1 = \lceil \lg(M+1) \rceil$. On the other hand, if $T \geq l(M) + 1$, then $\left\lceil \frac{l(M)+1}{T} \right\rceil = 1$, and we clearly have $T \leq 2^e$.

Now that our claim is proved, since $m_0 \geq M$ and $m_0 - (l(m_0) - 1)$ is a multiple of T we indeed have $y_m = y_{l(m)-1}$ for $m = m_0$. To finish proving the first part of the proposition, we must show that $\frac{3}{2}l(m_0) \leq m_0$ (the other inequality being trivial), or equivalently, keeping our notations above, that

$$T\left\lceil \frac{l(M)+1}{T} \right\rceil - 1 \geq 2^{e-1}.$$

Now clearly the left hand side is greater than or equal to $T - 1$, and on the other hand $2^{\lceil \lg T \rceil - 1} \leq 2^{\lg T} - 1 = T - 1$. Furthermore, the left hand side is also greater than or equal to $l(M) = 2^{\lfloor \lg M \rfloor}$, but one sees easily that $2^{\lceil \lg(M+1) \rceil - 1} = 2^{\lfloor \lg M \rfloor}$, thus showing the first part of the proposition. The proof of the second part (that is, the claim that m_0 is indeed the smallest) is similar (i.e. not illuminating) and is left to the reader. □

Using this proposition, we can decrease the number of comparisons in Brent's method since it will not be necessary to do anything (apart from a function evaluation) while m is between 2^e and $\frac{3}{2}2^e$.

8.5.3 Brent's Modified Algorithm

We temporarily return to our problem of factoring N. We must first explain how to choose f and x_0. The choice of x_0 seems to be quite irrelevant for the efficiency of the method. On the other hand, one must choose f carefully. In order to minimize the number of operations, we will want to take for f a polynomial of small degree. It is intuitively clear (and easy to prove) that linear polynomials f will not be random and hence give bad results. The quadratic polynomials on the other hand seem in practice to work pretty well, as long as we avoid special cases. The fastest to compute are the polynomials of the form $f(x) = x^2 + c$. Possible choices for c are $c = 1$ or $c = -1$. On the other hand $c = 0$ should, of course, be avoided. We must also avoid $c = -2$ since the recursion $x_{k+1} = x_k^2 - 2$ becomes trivial if one sets $x_k = u_k + 1/u_k$.

As already explained in Section 8.5.1, the "comparisons" $y_{k+t} = y_k$ are done by computing $(x_{k+t} - x_k, N)$. Now, even though we have studied efficient methods for GCD computation, such a computation is slow compared to a simple multiplication. Hence, instead of computing the GCD's each time, we batch them up by groups of 20 (say) by multiplying modulo N, and then do a single GCD instead of 20. If the result is equal to 1 (as will unfortunately usually be the case) then all the GCD's were equal to 1. If on the other hand it is non-trivial, we can backtrack if necessary.

The results and discussion above lead to the following algorithm.

Algorithm 8.5.2 (Pollard ρ). Given a composite integer N, this algorithm tries to find a non-trivial factor of N.

1. [Initialize] Set $y \leftarrow 2$, $x \leftarrow 2$, $x_1 \leftarrow 2$, $k \leftarrow 1$, $l \leftarrow 1$, $P \leftarrow 1$, $c \leftarrow 0$.
2. [Accumulate product] Set $x \leftarrow x^2 + 1 \bmod N$, $P \leftarrow P \cdot (x_1 - x) \bmod N$ and $c \leftarrow c + 1$. (We now have $m = 2l - k$, $l = l(m)$, $x = x_m$, $x_1 = x_{l(m)-1}$.) If $c = 20$, compute $g \leftarrow (P, N)$, then if $g > 1$ go to step 4 else set $y \leftarrow x$ and $c \leftarrow 0$.
3. [Advance] Set $k \leftarrow k - 1$. If $k \neq 0$ go to step 2. Otherwise, compute $g \leftarrow (P, N)$. If $g > 1$ go to step 4 else set $x_1 \leftarrow x$, $k \leftarrow l$, $l \leftarrow 2l$, then repeat k times $x \leftarrow x^2 + 1 \bmod N$, then set $y \leftarrow x$, $c \leftarrow 0$ and go to step 2.
4. [Backtrack] (Here we know that a factor of N has been found, maybe equal to N). Repeat $y \leftarrow y^2 + 1 \bmod N$, $g \leftarrow (x_1 - y, N)$ until $g > 1$ (this must occur). If $g < N$ output g, otherwise output a message saying that the algorithm fails. Terminate the algorithm.

Note that the algorithm may fail (indicating that the period modulo the different prime factors of N is essentially the same). In that case, do *not* start with another value of x_0, but rather with another polynomial, for example $x^2 - 1$ or $x^2 + 3$.

This algorithm has been further improved by P. Montgomery ([Mon2]) and R. Brent ([Bre2]).

8.5.4 Analysis of the Algorithm

As has already been said, it is not known how to analyze the above algorithms without assuming that f is a random map. Hence the analysis that we give is in fact an analysis of the iterates of a random map from a finite set E of cardinality p into itself. We also point out that some of the arguments given here are not rigorous but can be made so. We have given very few detailed analysis of algorithms in this book, but we make an exception here because the mathematics involved are quite pretty and the proofs short.

Call $P(M,T)$ the probability that a sequence of iterates y_m has preperiod M and period T. Then $y_0, \ldots, y_{M+T-1}$ are all distinct, and $y_{M+T} = y_M$. Hence we obtain

$$P(M,T) = \frac{1}{p} \prod_{1 \le k < M+T} \left(1 - \frac{k}{p}\right).$$

Now we will want to compute the asymptotic behavior as $p \to \infty$ of the average of certain functions over all maps f, i.e. of sums of the form

$$S = \sum_{M,T} P(M,T) g(M,T).$$

Now if we set $M = \mu\sqrt{p}$ and $T = \lambda\sqrt{p}$, we have

$$\begin{aligned} \ln(p \cdot P(M,T)) &= \sum_{k < (\lambda+\mu)\sqrt{p}} \ln\left(1 - \frac{k}{p}\right) = \sum_{k < (\lambda+\mu)\sqrt{p}} \left(-\frac{k}{p} + O\left(\frac{k^2}{p^2}\right)\right) \\ &= -\frac{(\lambda+\mu)^2}{2} + O\left(\frac{\lambda+\mu}{\sqrt{p}}\right) + O\left(\frac{(\lambda+\mu)^3}{\sqrt{p}}\right). \end{aligned}$$

Hence the limiting distribution of $P(M,L)dM\,dL$ is

$$\frac{1}{p} e^{-(\lambda+\mu)^2/2} \sqrt{p}\, d\mu \sqrt{p}\, d\lambda = e^{-(\lambda+\mu)^2/2} d\mu\, d\lambda,$$

so our sum S is asymptotic to

$$\int_0^\infty \int_0^\infty g(\mu\sqrt{p}, \lambda\sqrt{p}) e^{-(\lambda+\mu)^2/2} d\mu\, d\lambda. \qquad (*)$$

As a first application, let us compute the asymptotic behavior of the average of the period T.

Proposition 8.5.3. *As $p \to \infty$, the average of T is asymptotic to*

$$\sqrt{\frac{\pi p}{8}}.$$

Proof. Using $(*)$, we see that the average of T is asymptotic to

$$\sqrt{p}\int_0^\infty\int_0^\infty ye^{-(x+y)^2/2}dx\,dy.$$

By symmetry, this is equal to one half of the integral with $x+y$ instead of y, and this is easily computed and gives the proposition. □

Now we need to obtain the average of the other quantities entering into the expression for m_0 given in Proposition 8.5.1. Note that

$$T\left\lceil\frac{l(M)+1}{T}\right\rceil = T\left\lfloor\frac{l(M)}{T}\right\rfloor + T.$$

We then have

Proposition 8.5.4. *As $p\longrightarrow\infty$, the average of $T\left\lfloor\frac{l(M)}{T}\right\rfloor$ is asymptotic to*

$$\left(\frac{\ln\pi-\gamma}{2\ln 2}\right)\sqrt{\frac{\pi p}{8}}$$

where $\gamma = 0.57721\ldots$ is Euler's constant.

Proof. The proof is rather long, so we only sketch the main steps. Using $(*)$, the average of the quantity that we want to compute is asymptotic to

$$S=\int_0^\infty\int_0^\infty y\sqrt{p}\left\lfloor\frac{2^{\lfloor\lg(x\sqrt{p})\rfloor}}{y\sqrt{p}}\right\rfloor e^{-(x+y)^2/2}dx\,dy.$$

By splitting up the integral into pieces where the floor is constant, it is then a simple matter to show that

$$S=\sqrt{p}\sum_{n=1}^\infty\int_0^\infty yF\left(\frac{1}{\sqrt{p}}2^{\lceil\lg(ny\sqrt{p})\rceil}+y\right)dy,$$

where $F(y)=\int_y^\infty e^{-t^2/2}\,dt$. Now we assume that if we replace $\lceil\lg(ny\sqrt{p})\rceil$ by $\lg(ny\sqrt{p})+u$, where u is a uniformly distributed variable between 0 and 1, then S will be replaced by a quantity which is asymptotic to S (this step can be rigorously justified), i.e.

$$S\sim\sqrt{p}\sum_{n=1}^\infty\int_0^1 du\int_0^\infty yF(2^u ny+y)\,dy.$$

Now using standard methods like integration by parts and power series expansions, we find

$$S \sim \sqrt{\frac{\pi p}{8}} \frac{G(1) - G(1/2)}{\ln 2},$$

where

$$G(x) = \sum_{k=2}^{\infty} (-1)^k \frac{k-1}{k} \zeta(k) x^k$$

and $\zeta(s)$ is the Riemann zeta function. Now from the Taylor series expansion of the logarithm of the gamma function near $x = 1$, we immediately see that

$$G(x) = x \frac{\Gamma'(x+1)}{\Gamma(x+1)} - \ln \Gamma(x+1),$$

and using the special values of the gamma function and its derivative, we obtain Proposition 8.5.4. □

In a similar way (also by using the trick with the variable u), we can prove:

Proposition 8.5.5. *As $p \to \infty$, the average of*

$$2^{\lceil \lg \max(M+1,T) \rceil}$$

is asymptotic to

$$\frac{3}{2 \ln 2} \sqrt{\frac{\pi p}{8}}.$$

Combining these three propositions, we obtain the following theorem.

Theorem 8.5.6. *As $p \to \infty$, the average number of function evaluations in Algorithm 8.5.2 is asymptotic to*

$$FE = \left(\frac{3 + \ln 4\pi - \gamma}{2 \ln 2} \right) \sqrt{\frac{\pi p}{8}} \approx 3.1225 \sqrt{p},$$

and the number of multiplications mod N (i.e. implicitly of GCD's) is asymptotic to

$$MM = \left(\frac{\ln 4\pi - \gamma}{2 \ln 2} \right) \sqrt{\frac{\pi p}{8}} \approx 0.8832 \sqrt{p}.$$

This terminates our analysis of the Pollard ρ algorithm. As an exercise, the reader can work out the asymptotics for the unmodified Brent method and for the Pollard-Floyd method of detecting periodicity.

8.6 Shanks's Class Group Method

Another $O(N^{1/4+\epsilon})$ method (and even $O(N^{1/5+\epsilon})$ if one assumes the GRH) is due to Shanks. It is a simple by-product of the computation of the class number of an imaginary quadratic field (see Section 5.4). Indeed, let $D = -N$ if $N \equiv 3 \pmod 4$, $D = -4N$ otherwise. If h is the class number of $\mathbb{Q}(\sqrt{D})$ and if N is composite, then it is known since Gauss that h must be even (this is the start of the theory of genera into which we will not go). Hence, there must be an element of order exactly equal to 2 in the class group. Such an element will be called an ambiguous element, or in terms of binary quadratic forms, a form whose square is equivalent to the unit form will be called an ambiguous form.

Clearly, (a, b, c) is ambiguous if and only if it is equivalent to its inverse $(a, -b, c)$, and if the form is reduced this means that we have three cases.

(1) Either $b = 0$, hence $D = -4ac$, so $N = ac$.
(2) Or $a = b$, hence $D = b(b - 4c)$, hence $N = (b/2)(2c - b/2)$ if b is even, $N = b(4c - b)$ if b is odd.
(3) Or finally $a = c$, hence $D = (b - 2a)(b + 2a)$ hence $N = (b/2 + a)(a - b/2)$ if b is even, $N = (2a - b)(b + 2a)$ if b is odd.

We see that each ambiguous form gives a factorization of N (and this is a one-to-one correspondence).

Hence, Shanks's factoring method is roughly as follows: after having computed the class number h, look for an ambiguous form. Such a form will give a factorization of N (which may be trivial). There must exist a form which gives a non-trivial factorization however, and in practice it is obtained very quickly.

There remains the problem of finding ambiguous forms. But this is easy and standard. Write $h = 2^t q$ with q odd. Take a form f at random (for example one of the prime forms f_p used in Algorithm 5.4.10) and compute $g = f^q$. Then g is in the 2-Sylow subgroup of the class group, and if g is not the unit form, there exists an exponent m such that $0 \leq m < t$ and such that g^{2^m} is an ambiguous form. This is identical in group-theoretic terms to the idea behind the Rabin-Miller compositeness test (Section 8.2 above).

We leave to the reader the details of the algorithm which can be found in Shanks's paper [Sha1], as well as remarks on what should be done when the trivial factorization is found too often.

8.7 Shanks's SQUFOF

Still another $O(N^{1/4+\epsilon})$ method, also due to Shanks, is the SQUFOF (SQUare FOrm Factorization) method. This method is very simple to implement and also has the big advantage of working exclusively with numbers which are at most $2\sqrt{N}$, hence essentially half of the digits of N. Therefore it is eminently practical and fast when one wants to factor numbers less than 10^{19}, even on a pocket calculator. This method is based upon the infrastructure of real quadratic fields which we discussed in Section 5.8, although little of that appears in the algorithm itself.

Let D be a positive discriminant chosen to be a small multiple of the number N that we want to factor (for example we could take $D = N$ if $N \equiv 1 \pmod 4$, $D = 4N$ otherwise). Without loss of generality, we may assume that if $D \equiv 0 \pmod 4$, then $D/4 \equiv 2$ or $3 \pmod 4$, since otherwise we may replace D by $D/4$, and furthermore we may assume that D/N is squarefree, up to a possible factor of 4.

As in Shanks's class group method seen in the preceding section, we are going to look for ambiguous forms of discriminant D. Since here D is positive, we must be careful with the definitions. Recall from Chapter 5 that we have defined composition of quadratic forms only modulo the action of Γ_∞. We will say that a form is ambiguous if its square is *equal* to the identity modulo the action of Γ_∞, and not simply equivalent to it. In other words, the square of $f = (a, b, c)$ as given by Definition 5.4.6 must be of the form $(1, b', c')$. Clearly this is equivalent to $a \mid b$. Hence, a will be a factor of D, so once again ambiguous forms give us factorizations of D. The notion of ambiguous form must not be confused with the weaker notion of form belonging to an ambiguous cycle (see Section 5.7) which simply means that its square is equivalent to the identity modulo the action of $\mathrm{PSL}_2(\mathbb{Z})$ and not only of Γ_∞, i.e. belongs to the principal cycle.

Now let $g = (a, b, c)$ be a reduced quadratic form of discriminant D such that $a \mid c$. We note that since g is reduced hence primitive, we must have $\gcd(a, b) = 1$. Using Definition 5.4.6, one obtains immediately that

$$g^2 = (a^2, b, c/a),$$

this form being of course not necessarily reduced. This suggests the following idea.

We start from the identity form and use the ρ reduction operator used at length in Chapter 5 to proceed along the principal cycle, and we look for a form $f = (A, B, C)$ such that A is a square (such a form will be called a *square form*). We will see in a moment how plausible it is to believe that we can find such a form. Assume for the moment that we have found one, and set $A = a^2$ and $g = (a, B, aC)$.

Now g may not be primitive. In that case let p be a prime dividing the coefficients of g. Then if $p = 2$ we have $4 \mid A$ and $2 \mid B$. Hence, $D \equiv B^2 \equiv$

0 or 4 (mod 16), contradicting $D/4 \equiv 2$ or 3 (mod 4) when $4 \mid D$. If $p > 2$, then $p^2 \mid D$ hence since D/N or $D/(4N)$ is squarefree, we have $p^2 \mid N$. Although this case is rare in practice, it could occur, so we must compute $\gcd(a, B)$, and if this is not equal to 1 it gives a non-trivial factor of N (in fact its square divides N), and we can start the factorization after removing this factor.

Therefore we may assume that g is primitive. It is then clear from the definition that $g^2 = f$, whence the name "square form" given to f.

Now we start from $g^{-1} = (a, -B, aC)$ (which may not be reduced) and proceed along its cycle by applying the ρ operator. Since g^2 lies on the principal cycle, the reduced forms equivalent to g^{-1} will be on an ambiguous cycle.

Now we have the following proposition.

Proposition 8.7.1. *Keeping the above notations, there exists an ambiguous form g_1 on the cycle of g^{-1} at exactly half the distance (measured with the δ function introduced in Chapter 5) of f from the unit form.*

Proof. We prove this in the language of ideals, using the correspondence between classes of forms modulo Γ_∞ and classes of ideals modulo multiplication by $\mathbb{Q}^*$ given in Section 5.2.

Let $\mathfrak{a}$ be a representative of the ideal class (modulo $\mathbb{Q}^*$) corresponding to the quadratic form $g = (a, B, aC)$. Then by assumption, $\mathfrak{a}^2 = \gamma\mathbb{Z}_K$ for some $\gamma \in K$ which is of positive norm since $A = a^2 > 0$, and hence, in particular, $\mathcal{N}(\gamma) = \mathcal{N}(\mathfrak{a})^2$. Set

$$\beta = \gamma + \mathcal{N}(\mathfrak{a}) \quad \text{and} \quad \mathfrak{b} = \beta^{-1}\mathfrak{a}.$$

(Note that if desired, we can choose $a > 0$ and $\mathfrak{a}$ to be the unique primitive integral ideal corresponding to g, and then $\mathcal{N}(\mathfrak{a}) = a$.)

If, as usual, σ denotes real conjugation in K, we have chosen β such that

$$\frac{\sigma(\beta)}{\beta} = \frac{\mathcal{N}(\mathfrak{a})}{\gamma} = \frac{\sigma(\gamma)}{\mathcal{N}(\mathfrak{a})}.$$

Although it is trivial to give β explicitly, the knowledgeable reader will recognize that the existence of such a β is guaranteed by Hilbert's Theorem 90.

Now I claim that the quadratic form corresponding to $\mathfrak{b}$ is the ambiguous form that we are looking for. First, using the equations given above, we have

$$\mathfrak{b}^2 = \beta^{-2}\mathfrak{a}^2 = \frac{\gamma}{\beta^2}\mathbb{Z}_K = \frac{\mathcal{N}(\mathfrak{a})}{\mathcal{N}(\beta)}\mathbb{Z}_K$$

so the ideal $\mathfrak{b}^2$ is indeed equivalent up to multiplication by an element of $\mathbb{Q}^*$ to the unit ideal, so if g_1 is the quadratic form corresponding to $\mathfrak{b}^{-1}$, it is ambiguous.

Second, we clearly have $\gamma/\sigma(\gamma) = (\beta/\sigma(\beta))^2$ hence

$$\delta(g_1, g^{-1}) = \frac{1}{2}\ln\left|\frac{\beta}{\sigma(\beta)}\right| = \frac{1}{4}\ln\left|\frac{\gamma}{\sigma(\gamma)}\right| = \frac{1}{2}\delta(\mathbf{1}, f)$$

thus proving the proposition. □

Using this proposition, we see that with approximately half the number of applications of the ρ operator that were necessary to go from the identity to f, we go back from g^{-1} to an ambiguous form. In fact, since we know the exact distance that we have to go, we could use a form of the powering algorithm to make this last step much faster.

Now there are two problems with this idea. First, some ambiguous forms will correspond to trivial factorizations of N. Second, we have no guarantee that we will find square forms other than the identity. This will for instance be the case when the principal cycle is very short.

For the first problem, we could simply go on along the principal cycle if a trivial factorization is found. This would however not be satisfactory since for each square form that we encounter which may correspond to a trivial factorization, we would have to go back half the distance starting from g^{-1} before noticing this.

A good solution proposed by Shanks is as follows. Assume for the moment that $D = N$ or $D = 4N$. We obtain trivial factorizations of N exactly when the ambiguous cycle on which g^{-1} lies is the principal cycle itself. Hence, $f = g^2$ will be a square form which is equal to the square of a form on the principal cycle. Since all the forms considered are reduced, this can happen only if $g = (a, b, c)$ with $a^2 < \sqrt{D}$, hence $|a| < D^{1/4}$, which is quite a rare occurrence. When such an a occurs, we store $|a|$ in a list of dubious numbers, which Shanks calls the *queue*. Note that the condition $|a| < D^{1/4}$ is a necessary, but in general not a sufficient condition for the form g to be on the principal cycle, hence we may be discarding some useful numbers. In practice, this has little importance.

Now when a square form (A, B, C) with $A = a^2$ is found, we check whether a is in the queue. If it is, we ignore it. Otherwise, we are certain that the corresponding square root g is not in the principal cycle. (Note that the distance of the identity to $f = g^2$ is equal to twice the distance of the identity to g. This means that if g was in the principal cycle, we would have encountered it *before* encountering f.) Hence, we get a non-trivial factorization of D. This may of course give the spurious factors occurring in D/N, in which case one must go on. In fact, one can in this case modify the queue so that these factorizations are also avoided.

The second problem is more basic: what guarantee do we have that we can find a square form different from the identity in the principal cycle? For example, when the length of the cycle is short, there are none. This is the case, for example, for numbers N of the form $N = a^2 + 4$ for a odd, where the length of the cycle is equal to 1.

There are two different and complementary answers to this question. First, a heuristic analysis of the algorithm shows that the average number of reduc-

tion steps necessary to obtain a useful square form is $O(N^{1/4})$ (no ϵ here). This is much shorter than the usual length of the period which is in general of the order of $O(N^{1/2})$, so we can reasonably hope to obtain a square form before hitting the end of the principal cycle.

Second, to avoid problems with the length of the period, it may be worthwhile to work simultaneously with two discriminants D which are multiples of N, for example N and $5N$ when $N \equiv 1 \pmod 4$, $3N$ and $4N$ when $N \equiv 3 \pmod 4$. It is highly unlikely that both discriminants will have short periods. In addition, although the average number of reduction steps needed is on the order of $N^{1/4}$, experiments show that there is a very large dispersion around the mean, some numbers being factored much more easily than others. This implies that by running simultaneously two discriminants, one may hope to gain a substantial factor on average, which would compensate for the fact that twice as much work must be done.

We now give the basic algorithm, i.e. using only $D = N$ if $N \equiv 1 \pmod 4$, $D = 4N$ otherwise, and not using the fact than once g is found we can go back much faster by keeping track of distances.

Algorithm 8.7.2 (Shanks's SQUFOF). Given an odd integer N, this algorithm tries to find a non-trivial factor of N.

1. [Is N prime?] Using Algorithm 8.2.2, check whether N is a probable prime. If it is, output a message to that effect and terminate the algorithm.
2. [Is N square?] Using Algorithm 1.7.3, test whether N is a square. If it is, let n be its square root (also given by the algorithm), output n and terminate the algorithm.
3. [Initializations] If $N \equiv 1 \pmod 4$, let $D \leftarrow N$, $d \leftarrow \lfloor \sqrt{D} \rfloor$, $b \leftarrow 2\lfloor (d-1)/2 \rfloor + 1$. Otherwise, let $D \leftarrow 4N$, $d \leftarrow \lfloor \sqrt{D} \rfloor$, $b \leftarrow 2\lfloor d/2 \rfloor$. Then set $f \leftarrow (1, b, (b^2 - D)/4)$, $Q \leftarrow \emptyset$ (Q is going to be our queue), $i \leftarrow 0$, $L \leftarrow \lfloor \sqrt{d} \rfloor$.
4. [Apply rho] Let $f = (A, B, C) \leftarrow \rho(f)$, where ρ is given by Definition 5.6.4, and set $i \leftarrow i + 1$. If i is odd, go to step 7.
5. [Squareform?] Using Algorithm 1.7.3, test whether A is a square. If it is, let a be the (positive) square root of A (which is also output by Algorithm 1.7.3) and if $a \notin Q$ go to step 8.
6. [Short period?] If $A = 1$, output a message saying that the algorithm ran through the i elements of the principal cycle without finding a non-trivial squareform, and terminate the algorithm.
7. [Fill queue and cycle] If $|A| \leq L$, set $Q \leftarrow Q \cup \{|A|\}$. Go to step 4.
8. [Initialize back-cycle] (Here we have found a non-trivial square form). Let $s \leftarrow \gcd(a, B, D)$. If $s > 1$, output s^2 as a factor of N and terminate the algorithm (or start again with N replaced by N/s^2). Otherwise, set $g \leftarrow (a, -B, aC)$. Apply ρ to g until g is reduced, and write $g = (a, b, c)$.
9. [Back-cycle] Let $b_1 \leftarrow b$ and $g = (a, b, c) \leftarrow \rho(g)$. If $b_1 \neq b$ go to step 9. Otherwise, output $|a|$ if a is odd, $|a/2|$ if a is even, and terminate the algorithm.

Some remarks are in order. First, it is essential that N be a composite number, otherwise the queue will fill up indefinitely without the algorithm finding a square form. Also, N must not be a square, otherwise we do not have a quadratic field to work with. This is the reason why steps 1 and 2 have been explicitly included.

Second, once these cases out of the way, experiment shows that the queue stays small. A storage capacity of 50 is certainly more than sufficient.

Third, during the back-cycle part of the algorithm, we need to test whether we hit upon our ambiguous form. To do this, we could use the necessary and sufficient condition that $a \mid b$. It is however a simple exercise (see Exercise 12) to show that this is equivalent to the condition $b_1 = b$ used in step 9.

Several improvements are possible to this basic algorithm, including those mentioned earlier. For example, the queue could be used to shorten the back-cycle length, starting at hg^{-1} instead of g^{-1}, where h is the form corresponding to the last element put in the queue. We will not dwell on this here.

One of the main reasons why SQUFOF is attractive is that it works exclusively with reduced quadratic forms (a, b, c) of discriminant at most a small multiple of N, hence such that a, b and c are of the order of $N^{1/2}$. This implies that the basic operations in SQUFOF are much faster than in the other factoring algorithms where operations on numbers of size N or N^2 must be performed. Of course, this is only a constant factor, but in practice it is very significant. Furthermore, the algorithm is extremely simple, so it can easily be implemented even on a 10-digit pocket calculator, and one can then factor numbers having up to 19 or 20 digits without any multi-precision arithmetic.

Unfortunately, SQUFOF is not sensitive to the size of the small prime factors of N, hence contrary to Pollard's rho method, cannot be used to cast out small primes. So if N has more than 25 digits, say, SQUFOF becomes completely useless, while Pollard rho still retains its value (although it is superseded by ECM for larger numbers, see Chapter 10).

8.8 The $p-1$-method

The last factoring method which we will study in this chapter is a little special for two reasons. First, it is not a general purpose factoring method, but a way to find quickly prime factors of N that may be very large, but which possess certain properties. Second, the idea behind the method has successfully been used in some of the most successful modern factoring method like the elliptic curve method (see Section 10.3). Hence it is important to understand this method at least as an introduction to Chapter 10.

8.8.1 The First Stage

We need a definition.

Definition 8.8.1. *Let B be a positive integer. A positive integer n will be said to be* B-smooth *if all the prime divisors of n are less than or equal to B. We will say that n is* B-powersmooth *if all prime powers dividing n are less than or equal to B.*

These notions of smoothness are quite natural in factoring methods, and we will see that they become essential in the modern methods. The idea behind the $p-1$ method is the following. Let p be a prime dividing the number N that we want to split (p is of course a priori unknown). Let $a > 1$ be an integer (which we can assume coprime to N by computing a GCD, otherwise N will have split). Then by Fermat's theorem, $a^{p-1} \equiv 1 \pmod{p}$. Now assume that $p-1$ is B-powersmooth for a certain B which is not too large. Then by definition $p-1$ divides the least common multiple of the numbers from 1 to B, which we will denote by $\mathrm{lcm}[1..B]$. Hence, $a^{\mathrm{lcm}[1..B]} \equiv 1 \pmod{p}$, which implies that

$$(a^{\mathrm{lcm}[1..B]} - 1, N) > 1.$$

As in the Pollard ρ method, if this is tested for increasing values of B, it is highly improbable that this GCD will be equal to N, hence we will have found a non-trivial divisor of N. This leads to the following algorithm, which in this form is due to Pollard.

Algorithm 8.8.2 ($p-1$ First Stage). Let N be a composite number, and B be an a priori chosen bound. This algorithm will try to find a non-trivial factor of N, and has a chance of succeeding only when there exists a prime factor p of N such that $p-1$ is B-powersmooth. We assume that we have precomputed a table $p[1], \ldots, p[k]$ of all the primes up to B.

1. [Initialize] Set $x \leftarrow 2$, $y \leftarrow x$, $c \leftarrow 0$, $i \leftarrow 0$, and $j \leftarrow i$.
2. [Next prime] Set $i \leftarrow i+1$. If $i > k$, compute $g \leftarrow (x-1, N)$. If $g = 1$ output a message saying that the algorithm has not succeeded in splitting N, and terminate, else set $i \leftarrow j$, $x \leftarrow y$ and go to step 5. Otherwise (i.e. if $i \leq k$), set $q \leftarrow p[i]$, $q_1 \leftarrow q$, $l \leftarrow \lfloor B/q \rfloor$.
3. [Compute power] While $q_1 \leq l$, set $q_1 \leftarrow q \cdot q_1$. Then, set $x \leftarrow x^{q_1} \bmod N$, $c \leftarrow c+1$ and if $c < 20$ go to step 2.
4. [Compute GCD] Set $g \leftarrow (x-1, N)$. If $g = 1$, set $c \leftarrow 0$, $j \leftarrow i$, $y \leftarrow x$ and go to step 2. Otherwise, set $i \leftarrow j$ and $x \leftarrow y$.
5. [Backtrack] Set $i \leftarrow i+1$, $q \leftarrow p[i]$ and $q_1 \leftarrow q$.
6. [Finished?] Set $x \leftarrow x^q \bmod N$, $g \leftarrow (x-1, N)$. If $g = 1$, set $q_1 \leftarrow q \cdot q_1$ and if $q_1 \leq B$, go to step 6, else go to step 5. Otherwise (i.e. if $g > 1$), if $g < N$ output g and terminate the algorithm. Finally, if $g = N$ (a rare occurrence), output that the algorithm has failed and terminate.

Note that this algorithm may fail for two completely different reasons. The first one, by far the most common, occurs in step 2, and comes because N does not have any prime divisor p such that $p-1$ is B-powersmooth. In fact, it proves this. The second reason why it may fail occurs in step 6, but this is extremely rare. This would mean that all the prime p divisors of N are found simultaneously. If this is the case, then this means that there certainly exists a p dividing N which is B-powersmooth. Hence, it may be worthwhile to try the algorithm with a different initial value of x, for example $x \leftarrow 3$ instead of $x \leftarrow 2$.

Even in this simple form, the behavior of the $p-1$ algorithm is quite impressive. Of course, it does not pretend to be a complete factoring algorithm (in fact when $N = (2p+1)(2q+1)$ where p, q, $2p+1$ and $2q+1$ are primes with p and q about the same size, the running time of the algorithm will in general be $O(N^{1/2+\epsilon})$ if we want to factor N completely, no better than trial division). On the other hand, it may succeed in finding very large factors of N, since it is not the size of the prime factors of N which influence the running time but rather the smoothness of the prime factors minus 1.

The size of B depends essentially on the time that one is willing to spend. It is however also strongly conditioned by the existence of a second stage to the algorithm as we shall see presently. Usual values of B which are used are, say, between 10^5 and 10^6.

8.8.2 The Second Stage

Now an important practical improvement to the $p-1$ algorithm (which one also uses in the modern methods using similar ideas) is the following. It may be too much to ask that there should exist a prime divisor p of N such that $p-1$ is B-powersmooth. It is more reasonable to ask that $p-1$ should be completely factored by trial division up to B. But this means that $p-1 = fq$, where f is B-smooth, and q is a prime which may be much larger than B (but not than B^2). For our purposes, we will slightly strengthen this condition and assume that N has a prime factor p such that $p-1 = fq$ where f is B_1-powersmooth and q is a prime such that $B_1 < q \leq B_2$, where B_1 is our old B, and B_2 is a much larger constant. We must explain how we are going to find such a p. Of course, $p-1$ is B_2-powersmooth so we could use the $p-1$ algorithm with B_1 replaced by B_2. This is however unrealistic since B_2 is much larger than B_1.

Now we have as usual

$$(a^{q\,\mathrm{lcm}[1..B_1]} - 1, N) > 1$$

and we will proceed as follows. At the end of the first stage (i.e. of Algorithm 8.8.2 above), we will have computed $b \leftarrow a^{\mathrm{lcm}[1..B_1]} \bmod N$. We store a table of the difference of primes from B_1 to B_2. Now these differences are small, and there will not be many of them. So we can quickly compute b^d for all possible

differences d, and obtain all the b^q by *multiplying* successively an initial power of b by these precomputed b^d. Hence, for each prime, we replace a powering operation by a simple multiplication, which is of course *much* faster, and this is why we can go much further. This leads to the following algorithm.

Algorithm 8.8.3 ($p-1$ with Stage 2). Let N be a composite number, and B_1 and B_2 be a priori chosen bounds. This algorithm will try to find a non-trivial factor of N, and has a chance of succeeding only when there exists a prime factor p of N such that $p-1$ is equal to a B_1-powersmooth number times a prime less than or equal to B_2. We assume that we have precomputed a table $p[1], \ldots, p[k_1]$ of all the primes up to B_1 and a table $d[1], \ldots, d[k_2]$ of the differences of the primes from B_1 to B_2, with $d[1] = p[k_1+1] - p[k_1]$, etc ...

1. [First stage] Using $B = B_1$, try to split N using Algorithm 8.8.2 (i.e. the first stage. If this succeeds, terminate the algorithm. Otherwise, we will have obtained a number x at the end of Algorithm 8.8.2, and we set $b \leftarrow x$, $c \leftarrow 0$, $P \leftarrow 1$, $i \leftarrow 0$, $j \leftarrow i$ and $y \leftarrow x$.
2. [Precomputations] For all values of the differences $d[i]$ (which are small and few in number), precompute and store $b^{d[i]}$. Set $x \leftarrow x^{p[k_1]}$.
3. [Advance] Set $i \leftarrow i+1$, $x \leftarrow x \cdot b^{d[i]}$ (using the precomputed value of $b^{d[i]}$), $P \leftarrow P \cdot (x-1)$, $c \leftarrow c+1$. If $i \geq k_2$, go to step 6. Otherwise, if $c < 20$, go to step 3.
4. [Compute GCD] Set $g \leftarrow (P, N)$. If $g = 1$, set $c \leftarrow 0$, $j \leftarrow i$, $y \leftarrow x$ and go to step 3.
5. [Backtrack] Set $i \leftarrow j$, $x \leftarrow y$. Then repeat $x \leftarrow x \cdot b^{d[i]}$, $i \leftarrow i+1$, $g \leftarrow (x-1, N)$ until $g > 1$ (this must occur). If $g < N$ output g and terminate the algorithm. Otherwise (i.e. if $g = N$, a rare occurrence), output that the algorithm has failed (or try again using $x \leftarrow 3$ instead of $x \leftarrow 2$ in the first step of Algorithm 8.8.2), and terminate.
6. [Failed?] Set $g \leftarrow (P, N)$. If $g = 1$, output that the algorithm has failed and terminate. Otherwise go to step 5.

In this form, the $p-1$ algorithm is much more efficient than using the first stage alone. Typical values which could be used are $B_1 = 2 \cdot 10^6$, $B_2 = 10^8$. See also [Mon2] and [Bre2] for further improvements.

8.8.3 Other Algorithms of the Same Type

The main drawback of the $p-1$ algorithm is that there is no reason for N to have a prime divisor p such that $p-1$ is smooth. As with the primality tests (see Section 8.3.2), we can also detect the primes p such that $p+1$ is smooth, or also p^2+p+1, p^2+1, p^2-p+1 (although since these numbers are much larger, their probability of being smooth for a given bound B is much smaller). We leave as an exercise for the reader (Exercise 13) to write an algorithm when $p+1$ is B-powersmooth.

We see that the number of available groups which give numbers of reasonable size (here $\mathbb{F}_p^*$ and $\mathbb{F}_{p^2}^*/\mathbb{F}_p^*$, which give $p-1$ and $p+1$ respectively) is very small (2) and this limits the usefulness of the method. The idea of the elliptic curve method (ECM) is to use the group of points of an elliptic curve over $\mathbb{F}_p$, which also has approximately p elements by Hasse's Theorem 7.1.8, and this will lead to a much better algorithm since we will have at our disposal a large number of groups of small size instead of only two. See Section 10.3 for details.

8.9 Exercises for Chapter 8

1. Show that an odd prime number p is a strong pseudo-prime in any base not divisible by p.
2. If N is the 46 digit composite number due to Arnault given in the text as an example of a strong pseudoprime to all prime bases $a \leq 31$, compute explicitly $a^{(N-1)/4} \bmod N$ for these a and show that -1 has at least 5 different square roots modulo N (showing clearly N that is not prime even without knowing its explicit factorization). From this remark, deduce a strengthening of the Rabin-Miller test which would not be passed for example by Arnault's number.
3. Show that if N is any odd integer, the congruence

$$a^{N-1} \equiv -1 \pmod{N}$$

is impossible. More generally, show that

$$a^k \equiv -1 \pmod{N}$$

implies that

$$v_2(k) \leq v_2\left(\frac{N-1}{2}\right).$$

The following four exercises are due to H. W. Lenstra.

4. Show that there are only a finite number of integers N such that for all $a \in \mathbb{Z}$ we have

$$a^{N+1} \equiv a \pmod{N},$$

and give the complete list.

5. Let N be a positive integer such that $2^N \equiv 1 \pmod{N}$. Show that $N = 1$.
6. Let a be a positive integer such that $a^4 + 4^a$ is a prime number. Show that $a = 1$.
7. Show that there exists infinitely many n for which at least one of $2^{2^n} + 1$ or $6^{2^n} + 1$ is composite.
8. Denote by F_k the k-th Fermat number, i.e. $F_k = 2^{2^k} + 1$.
   a) Show manually that F_k is prime for $0 \leq k \leq 4$ but that $641 \mid F_5$.
   b) Let $h > 1$ be an integer such that $h \equiv 1 \pmod{F_0F_1F_2F_3F_4}$. If $h2^n + 1$ is prime show that $32 \mid n$.
   c) Conclude that there exists an a such that if

$$h \equiv a \pmod{F_0F_1F_2F_3F_4F_5}$$

and $h > 1$, then for all n, $h2^n + 1$ is composite.

9. Let $N = 2^{2^k} + 1$ be a Fermat number. Prove that in this case Proposition 8.3.1 can be made more precise as follows: N is prime if and only if $3^{(N-1)/2} \equiv -1 \pmod{N}$ (use the quadratic reciprocity law).

10. Using implicitly the finite field $\mathbb{F}_{N^2}$, write a primality testing algorithm in the case where $N+1$ is completely factored, using a proposition similar to 8.3.1.

11. Using the algorithm developed in Exercise 10, show that the Mersenne number $N = 2^p - 1$ is prime if and only p is prime and (for $p \neq 2$) if the sequence defined by $u_0 = 4$ and $u_{k+1} = u_k^2 - 2 \bmod N$ satisfies $u_{p-2} = 0$ (this is called the Lucas-Lehmer test).

12. Let $g = (a, b, c)$ and $g_1 = (a_1, b_1, c_1) = \rho(g)$ be reduced forms with positive discriminant. Show that g_1 is an ambiguous form if and only if $b = b_1$.

13. The $p-1$-algorithm is based on the properties of the finite field $\mathbb{F}_p$. Using instead the field $\mathbb{F}_{p^2}$, develop a $p+1$-factoring algorithm for use when a prime factor p of N is such that $p+1$ is B-powersmooth for some reasonable bound B.

14. Let N be a number to be factored. Assume that after one of the factoring algorithms seen in this chapter we have found a number a such that $d = \gcd(N, a)$ satisfies $1 < d < N$ hence gives a non-trivial divisor of N. Write an algorithm which extracts as much information as possible from this divisor d, i.e. which finds N_1 and N_2 such that $N = N_1N_2$, $\gcd(N_1, N_2) = 1$ and $d \mid N_1$.

Chapter 9

Modern Primality Tests

In Section 8.3, we studied various primality tests, essentially the $N-1$ test, and saw that they require knowing the factorization of $N-1$ (or $N+1, \ldots$), which are large numbers. Even though only partial factorizations are needed, the tests of Section 8.3 become impractical as soon as N has more than 100 digits, say. A breakthrough was made in 1980 by Adleman, Pomerance and Rumely, that enabled testing the primality of much larger numbers. The APR test was further simplified and improved by H. W. Lenstra and the author, and the resulting APRCL test was implemented in 1981 by A. K. Lenstra and the author, with the help of D. Winter. It is now possible to prove the primality of numbers with 1000 decimal digits in a not too unreasonable amount of time. The running time of this algorithm is $O((\ln N)^{C \ln\ln\ln N})$ for a suitable constant C. This is almost a polynomial time algorithm since for all practical purposes the function $\ln\ln\ln N$ acts like a constant. (Note that the practical version of the algorithm is probabilistic, but that there exists a non-probabilistic but less practical version.)

We will describe the algorithm in Section 9.1, without giving all the the implementation tricks. The reader will find a detailed description of this algorithm and its implementation in [Coh-Len2], [Coh-Len3] and [Bos-Hul].

In 1986, another primality testing algorithm was invented, first for theoretical purposes by Goldwasser and Kilian, and then considerably modified so as to obtain a practical algorithm by Atkin. This algorithm has been implemented by Atkin and Morain, and is also practical for numbers having up to 1000 digits. The expected running time of this algorithm is $O(\ln^6 N)$, hence is polynomial time, but this is only on average since for some numbers the running time could be much larger. A totally non-practical version using a higher dimensional analog of this test has been given by Adleman and Huang, and they can prove that their test is polynomial time. In other words, they prove the following theorem ([Adl-Hua]).

Theorem 9.1. *There exists a probabilistic polynomial time algorithm which can prove or disprove that a given number N is prime.*

Their proof is pretty but very complex, and this theorem is one of the major achievements of theoretical algorithmic number theory.

We will describe Atkin's practical primality test in Section 9.2, and we refer to [Atk-Mor] and to [Mor2] for implementation details.

9.1 The Jacobi Sum Test

The idea of the APRCL method is to test Fermat-type congruences in higher degree number fields, and more precisely in certain well chosen cyclotomic fields. We need a few results about group rings in this context.

9.1.1 Group Rings of Cyclotomic Extensions

Recall first the following definitions and results about cyclotomic fields (see [Was]).

Definition 9.1.1. *If n is a positive integer, the n-th* cyclotomic field *is the number field $\mathbb{Q}(\zeta_n)$, where ζ_n is a primitive n-th root of unity, for example $\zeta_n = e^{2i\pi/n}$.*

Proposition 9.1.2. *Let $K = \mathbb{Q}(\zeta_n)$ be the n-th cyclotomic field.*

(1) *The extension $K/\mathbb{Q}$ is a Galois extension, with Abelian Galois group given by*

$$G = \mathrm{Gal}(K/\mathbb{Q}) = \{\sigma_a,\ (a,n) = 1,\ \text{where } \sigma_a(\zeta_n) = \zeta_n^a\}.$$

In particular, the degree of $K/\mathbb{Q}$ is $\phi(n)$, where ϕ is Euler's phi function.

(2) *The ring of integers of K is $\mathbb{Z}_K = \mathbb{Z}[\zeta_n]$.*

We now come to the definition of a group ring. We could of course bypass this definition, but the notations would become very cumbersome.

Definition 9.1.3. *Let G be any finite group. The* group ring $\mathbb{Z}[G]$ *is the set of maps (not necessarily homomorphisms) from G to $\mathbb{Z}$ with the following two operations. If f_1 and f_2 are in $\mathbb{Z}[G]$, we naturally define*

$$(f_1 + f_2)(\sigma) = f_1(\sigma) + f_2(\sigma)$$

for all $\sigma \in G$. The multiplication law is more subtle, and is defined by

$$f_1 \cdot f_2(\sigma) = \sum_{\tau \in G} f_1(\tau) f_2(\tau^{-1}\sigma).$$

The name group ring is justified by the easily checked fact that the above operations do give a ring structure to $\mathbb{Z}[G]$. If for $f \in \mathbb{Z}[G]$, we set formally

$$f = \sum_{\sigma \in G} f(\sigma)[\sigma],$$

where $[\sigma]$ is just a notation, then it is easy to see that addition and multiplication become natural $\mathbb{Z}$-algebra laws, if we set, as is natural, $[\sigma_1]\cdot[\sigma_2]=[\sigma_1\sigma_2]$. This is the notation which we will use. Note also that although we have only defined group rings $\mathbb{Z}[G]$ for finite groups G, it is easy to extend this to infinite groups by requiring that all but a finite number of images of the maps be equal to 0 (in order to have finite sums).

We can consider $\mathbb{Z}$ as a subring of $\mathbb{Z}[G]$ by identifying n with $n[1]$, where 1 is the unit element of G, and we will use this identification from now on.

We now specialize to the situation where $G=\mathrm{Gal}(K/\mathbb{Q})$ for a number field K Galois over $\mathbb{Q}$, and in particular to the case where K is a cyclotomic field. By definition, the group G acts on K, and also on all objects naturally associated to K: the unit group, the class group, etc ... One can extend this action of G in a natural way to an action of $\mathbb{Z}[G]$ in the following way. If $f\in\mathbb{Z}[G]$ and $x\in K$, then we set

$$f(x)=\prod_{\sigma\in G}\sigma(x)^{f(\sigma)}.$$

In the expanded form where we write $f=\sum_{\sigma\in G}n_\sigma[\sigma]$, one sees immediately that this corresponds to a *multiplicative* extension of the action of G, and suggests using the notation x^f instead of $f(x)$ so that

$$x^f=\prod_{\sigma\in G}\sigma(x)^{n_\sigma}.$$

Indeed, it is easy to check the following properties (x, x_1 and x_2 are in K and f, f_1 and f_2 are in $\mathbb{Z}[G]$):

(1) $x^{f_1+f_2}=x^{f_1}\cdot x^{f_2}$.
(2) $x^{f_1\cdot f_2}=\left(x^{f_1}\right)^{f_2}=\left(x^{f_2}\right)^{f_1}$.
(3) $(x_1+x_2)^f=x_1^f+x_2^f$
(4) $(x_1x_2)^f=x_1^fx_2^f$

We now fix a prime number p and an integer k, and consider the n-th cyclotomic field K, where $n=p^k$. Let G be its Galois group, which is the set of all σ_a for $a\in(\mathbb{Z}/n\mathbb{Z})^*$ by Proposition 9.1.2. Since it is Abelian, the group ring $\mathbb{Z}[G]$ is a commutative ring. Set

$$\mathfrak{p}=\{f\in\mathbb{Z}[G]\ /\ \zeta_p^f=1\},$$

where $\zeta_p=e^{2i\pi/p}$ is a primitive p^{th} root (not p^k) of unity. Then one checks immediately that $\mathfrak{p}$ is an ideal of $\mathbb{Z}[G]$. In fact, if $f=\sum_{a\in(\mathbb{Z}/n\mathbb{Z})^*}n_a[\sigma_a]$, then $f\in\mathfrak{p}$ if and only if $\sum_{a\in(\mathbb{Z}/n\mathbb{Z})^*}an_a\equiv 0\pmod p$. This shows that the number of cosets of $\mathbb{Z}[G]$ modulo $\mathfrak{p}$ is equal to p (the number of different incongruent sums $\sum an_a$ modulo p), hence that $\mathfrak{p}$ is in fact a *prime* ideal of degree one (i.e. of norm equal to p). Clearly, it is generated over $\mathbb{Z}$ by p (i.e. $p[1]$) and all the $a-[\sigma_a]$.

9.1.2 Characters, Gauss Sums and Jacobi Sums

Recall that a *character* (more precisely a Dirichlet character) χ modulo q is a group homomorphism from $(\mathbb{Z}/q\mathbb{Z})^*$ to $\mathbb{C}^*$ for some integer q. This can be naturally extended to a multiplicative map from $(\mathbb{Z}/q\mathbb{Z})$ to $\mathbb{C}$ by setting $\chi(x) = 0$ if $x \notin (\mathbb{Z}/q\mathbb{Z})^*$. It can then be lifted to a map from $\mathbb{Z}$ to $\mathbb{C}$, which by abuse of notation we will still denote by χ. The set of characters modulo q forms a group, and for instance using Section 1.4.1 one can easily show that this group is (non-canonically) isomorphic to $(\mathbb{Z}/q\mathbb{Z})^*$, and in particular has $\phi(q)$ elements. The unit element of this group is the character χ_0 such that $\chi_0(x) = 1$ if $(x, q) = 1$ and 0 otherwise.

Proposition 9.1.4. *Let χ be a character different from χ_0. Then*

$$\sum_{x \in (\mathbb{Z}/q\mathbb{Z})^*} \chi(x) = 0.$$

Dually, if $x \not\equiv 1 \pmod{q}$, then

$$\sum_{\chi} \chi(x) = 0,$$

where the sum is over all characters modulo q.

Proof. Since $\chi \neq \chi_0$, there exists a number a coprime to q such that $\chi(a) \neq 1$. Set $S = \sum_x \chi(x)$. Since χ is multiplicative we have $\chi(a)S = \sum_x \chi(ax)$. Since a is coprime to q and hence invertible modulo q, the map $x \mapsto ax$ is a bijection of $(\mathbb{Z}/q\mathbb{Z})^*$ onto itself. It follows that $\chi(a)S = \sum_y \chi(y) = S$, and since $\chi(a) \neq 1$, this shows that $S = 0$ as claimed. The second part of the proposition is proved in the same way using the existence of a character χ_1 such that $\chi_1(x) \neq 1$ when $x \not\equiv 1 \pmod{q}$. □

The *order* of a character χ is the smallest positive n such that $\chi(a)^n = 1$ for all integers a prime to q, in other words it is the order of χ considered as an element of the group of characters modulo q.

Definition 9.1.5.

(1) *Let χ be a character modulo q. The* Gauss sum *$\tau(\chi)$ is defined by*

$$\tau(\chi) = \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^*} \chi(x)\zeta_q^x,$$

where as usual $\zeta_q = e^{2i\pi/q}$.

(2) *Let χ_1 and χ_2 be two characters modulo q. The* Jacobi sum *$j(\chi_1, \chi_2)$ is defined by*

$$j(\chi_1, \chi_2) = \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^*} \chi_1(x)\chi_2(1-x).$$

Note that since we have extended characters by 0, we can replace $(\mathbb{Z}/q\mathbb{Z})^*$ by $\mathbb{Z}/q\mathbb{Z}$, and also that in the definition of Jacobi sums, one could exclude $x = 1$ which contributes 0 to the sum.

From the definitions, it is clear that if χ is a character modulo q of order n (hence $n \mid \phi(q)$), then

$$\tau(\chi) \in \mathbb{Z}[\zeta_n, \zeta_q],$$

while if χ_1 and χ_2 are two characters modulo q of order dividing n, then

$$j(\chi_1, \chi_2) \in \mathbb{Z}[\zeta_n].$$

This will in general be a much simpler ring than $\mathbb{Z}[\zeta_n, \zeta_q]$, and this observation will be important in the test.

The basic results about Gauss sums and Jacobi sums that we will need are summarized in the following proposition. Note that we assume that q is a prime, which makes things a little simpler.

Proposition 9.1.6.

(1) *Let $\chi \neq \chi_0$ be a character modulo a prime q. Then*

$$\tau(\chi)\tau(\bar{\chi}) = \chi(-1)q \quad \textit{and} \quad |\tau(\chi)| = \sqrt{q}\,.$$

(2) *Let χ_1 and χ_2 be two characters modulo q such that $\chi_1\chi_2 \neq \chi_0$. Then*

$$j(\chi_1, \chi_2) = \frac{\tau(\chi_1)\tau(\chi_2)}{\tau(\chi_1\chi_2)}.$$

Proof. To simplify notations, except if explicitly stated otherwise, the summations will always be over $(\mathbb{Z}/q\mathbb{Z})^*$, and we abbreviate ζ_q to ζ. We have:

$$\tau(\chi)\tau(\bar{\chi}) = \sum_x \chi(x)\zeta^x \sum_y \bar{\chi}(y)\zeta^y = \sum_t \chi(t) \sum_y \chi(y)\bar{\chi}(y)\zeta^{y(1+t)},$$

by setting $x = ty$. Since $\chi(y)\bar{\chi}(y) = 1$, the inner sum is simply a sum of powers of ζ, and since q is prime, is a geometric series whose sum is equal to -1 if $1 + t \neq 0$ and to $q - 1$ otherwise. Hence, our product is equal to

$$-\sum_{t \neq -1} \chi(t) + (q-1)\chi(-1) = q\chi(-1) - \sum_t \chi(t) = q\chi(-1)$$

by Proposition 9.1.4. Finally, note that

$$\overline{\tau(\chi)} = \sum_x \bar{\chi}(x)\zeta^{-x} = \sum_x \bar{\chi}(-x)\zeta^x = \chi(-1)\tau(\bar{\chi}),$$

and the first part of the proposition is proved.

The second part is proved analogously. We have

$$\tau(\chi_1)\tau(\chi_2) = \sum_x \sum_y \chi_1(x)\chi_2(y)\zeta^{x+y} = \sum_t \sum_y \chi_1(t)\chi_1\chi_2(y)\zeta^{y(1+t)}$$

by setting $x = ty$. Now by setting $x = ay$ it is clear that for any $\chi \neq \chi_0$ we have

$$\sum_y \chi(y)\zeta^{ay} = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{q} \\ \bar{\chi}(a)\tau(\chi) & \text{otherwise.} \end{cases}$$

Hence, since $\chi_1\chi_2 \neq \chi_0$, we have

$$\tau(\chi_1)\tau(\chi_2) = \tau(\chi_1\chi_2) \sum_{t \neq -1} \chi_1(t)\overline{\chi_1\chi_2}(1+t) = \tau(\chi_1\chi_2) \sum_u \chi_1(u)\chi_2(1-u)$$

if we set $u = t/(1+t)$ which sends bijectively $(\mathbb{Z}/q\mathbb{Z}) \setminus \{0, -1\}$ onto $(\mathbb{Z}/q\mathbb{Z}) \setminus \{0, 1\}$, proving the identity. □

9.1.3 The Basic Test

We now come back to our basic purpose, i.e. testing the primality of a number N. It is assumed that N has already passed the Rabin-Miller test 8.2.2, so that it is highly improbable that N is composite. The aim is to *prove* that N is prime.

In this section, we fix a prime p and a character χ of order p^k modulo a prime q (hence with $p^k \mid (q-1)$). We can of course assume that N is prime to p and q. We set for simplicity $n = p^k$, and denote by $\langle\zeta_n\rangle$ the group of n-th roots of unity, which is generated by ζ_n. We shall use a modified version of Fermat's theorem as follows.

Proposition 9.1.7. *Let $\beta \in \mathbb{Z}[G]$. Then if N is prime, there exists $\eta(\chi) \in \langle\zeta_n\rangle$ such that*

$$\tau(\chi)^{\beta(N-\sigma_N)} \equiv \eta(\chi)^{-\beta N} \pmod{N}, \qquad (*_\beta)$$

where in fact $\eta(\chi) = \chi(N)$.

Note that we consider $\mathbb{Z}[G]$ as acting not only on $\mathbb{Q}(\zeta_n)$ but also on $\mathbb{Q}(\zeta_n, \zeta_q)$, the action being trivial on ζ_q. Note also that the congruences modulo N are in fact modulo $N\mathbb{Z}[\zeta_n, \zeta_q]$.

Proof. We know that in characteristic N, $(\sum a_k)^N = \sum a_k^N$ since the binomial coefficients $\binom{N}{i}$ are divisible by N if $0 < i < N$. Hence,

$$\tau(\chi)^N \equiv \sum_x \chi(x)^N \zeta_q^{Nx} \equiv \sum_x \chi(N^{-1}x)^N \zeta_q^x \equiv \chi(N)^{-N}\tau(\chi^N) \pmod{N}$$

and the proposition follows since $\tau(\chi^N) = \tau(\chi)^{\sigma_N}$ by definition of σ_N. Note that $\tau(\chi)$ is also coprime to N since by Proposition 9.1.6, $\tau(\chi)\overline{\tau(\chi)} = q$ is coprime to N. □

This proposition is a generalization of Fermat's theorem since one checks immediately that if we take $n = p = 2$ and $\beta = 1$, the proposition is equivalent to the statement $q^{(N-1)/2} \equiv \pm 1 \pmod{N}$. What we are now going to prove is in essence that if, conversely, condition $(*_\beta)$ is satisfied for a number of characters χ (with different p^k and q), then we can easily finish the proof that N is prime. First, we prove the following

Lemma 9.1.8. *Let N be any integer, and assume that $(*_\beta)$ is satisfied. Then*

(1) *For all $i > 0$*

$$\tau(\chi)^{\beta\left(N^i - \sigma_{N^i}\right)} \equiv \eta(\chi)^{-\beta i N^i} \pmod{N}.$$

(2)

$$\tau(\chi)^{\beta\left(N^{(p-1)p^{k-1}} - 1\right)} \equiv \eta(\chi)^{\beta p^{k-1}} \pmod{N}.$$

(3) *If r is prime and coprime to p and q then*

$$\tau(\chi)^{\left(r^{(p-1)p^{k-1}} - 1\right)} \equiv \chi(r)^{p^{k-1}} \pmod{r}.$$

Proof. Assertion (1) follows from $(*_\beta)$ by induction on i using the identity

$$N^{i+1} - \sigma_{N^{i+1}} = N^i\left(N - \sigma_N\right) + \sigma_N\left(N^i - \sigma_{N^i}\right)$$

and $\eta(\chi)^{\sigma_N} = \eta(\chi)^N$ since $\eta(\chi) \in \langle\zeta_n\rangle$. For (2) we apply the first assertion to $i = (p-1)p^{k-1}$ and use Euler's Theorem 1.4.2 which tells us that

$$N^{(p-1)p^{k-1}} \equiv 1 \pmod{p^k}.$$

The last assertion follows immediately since Proposition 9.1.7 tells us that $(*_\beta)$ is satisfied for a prime number r with $\beta = 1$ and $\eta(\chi) = \chi(r)$. □

We now introduce a condition which will be crucial to all our future work. We will show that this condition is a consequence of $(*_\beta)$ conditions for suitable characters χ. This means that it will have a similar nature to the Fermat tests, but it is much more convenient to isolate it from the rest of the tests.

Definition 9.1.9. *We say that condition $\mathcal{L}_p$ is satisfied (with respect to N of course) if for all prime divisors r of N and all integers $a > 0$ we can find an integer $l_p(r, a)$ such that*

$$r^{p-1} \equiv N^{(p-1)l_p(r,a)} \pmod{p^a}.$$

Note that if N is prime this condition is trivially satisfied with $l_p(r,a) = 1$. We will see later that this condition is not as difficult as it looks and that it can easily be checked. For the moment, let us see what consequences we can deduce from it. Note first that if $l_p(r,a)$ exists for all primes r dividing N, it exists by additivity for every divisor r of N.

Note also that condition $\mathcal{L}_p$ is more nicely stated in p-adic terms, but we will stay with the present definition. One consequence of this fact which we will use (and prove later) is the following result.

Lemma 9.1.10. *Let $u = v_p\left(N^{p-1}-1\right)$ if $p \geq 3$, $u = v_2\left(N^2-1\right)$ if $p = 2$. Then for $a \geq b \geq u$ we have*

$$l_p(r,a) \equiv l_p(r,b) \pmod{p^{b-u}}.$$

The main consequence of condition $\mathcal{L}_p$ which we need is the following.

Proposition 9.1.11. *Assume that condition $\mathcal{L}_p$ is satisfied.*

(1) *If χ satisfies $(*_\beta)$ for some $\beta \notin \mathfrak{p}$, then for all sufficiently large a and all $r \mid N$ we have*

$$\chi(r) = \chi(N)^{l_p(r,a)} \quad \text{and} \quad \eta(\chi) = \chi(N).$$

(2) *If ψ is a character modulo a power of p and of order a power of p, then we also have*

$$\psi(r) = \psi(N)^{l_p(r,a)}$$

for sufficiently large a.

Proof. Set for simplicity $x = \tau(\chi)^\beta$. From the first part of Lemma 9.1.8 we have

$$x^{N^{(p-1)p^k}-1} \equiv 1 \pmod{N}.$$

Set $N^{(p-1)p^k} - 1 = p^e N_1$ with $p \nmid N_1$. Set $\ell = l_p(r, \max(e, k+u))$, where u is as in Lemma 9.1.10. Then again using the first part of Lemma 9.1.8 we have

$$\begin{aligned} x^{N^{(p-1)\ell}} &\equiv \eta(\chi)^{-\beta(p-1)\ell N^{(p-1)\ell}} \tau\left(\chi^{N^{(p-1)\ell}}\right)^\beta \pmod{N} \\ &\equiv \eta(\chi)^{-\beta(p-1)\ell r^{p-1}} \tau\left(\chi^{r^{p-1}}\right)^\beta \pmod{N} \end{aligned}$$

since $\eta(\chi)$ and χ are of order dividing p^k. If r is a prime divisor of N, we have by Proposition 9.1.7

$$x^{r^{p-1}} \equiv \chi(r)^{-\beta(p-1)r^{p-1}} \tau\left(\chi^{r^{p-1}}\right)^{\beta} \pmod{r}$$

hence, since $\tau\left(\chi^{r^{p-1}}\right)^{\beta}$ is invertible modulo r by Proposition 9.1.6, we obtain finally

$$x^{\left(N^{(p-1)\ell}-r^{p-1}\right)} \equiv \zeta^{\beta(p-1)r^{p-1}} \pmod{r} \quad \text{with } \zeta = \chi(r)\eta(\chi)^{-\ell}.$$

Now from our choice of ℓ, we have $N^{(p-1)\ell} \equiv r^{p-1} \pmod{p^e}$, hence

$$N_1\left(N^{(p-1)\ell} - r^{p-1}\right) \equiv 0 \pmod{N^{(p-1)p^k} - 1}.$$

So if we combine this with our preceding congruences we obtain

$$x^{N_1\left(N^{(p-1)\ell}-r^{p-1}\right)} \equiv 1 \equiv \zeta^{N_1\beta(p-1)r^{p-1}} \pmod{r}.$$

Now we trivially have $N_1\beta(p-1)r^{p-1} \notin \mathfrak{p}$ since $\mathfrak{p}$ is a prime ideal and none of the factors belong to $\mathfrak{p}$. Since ζ is a p^k-th root of unity, the definition of $\mathfrak{p}$ implies that it must be equal to 1, i.e. that

$$\chi(r) = \eta(\chi)^{\ell} = \eta(\chi)^{l_p(r,a)}$$

for a sufficiently large, and for all prime r dividing N (by Lemma 9.1.10 and our choice of ℓ). By additivity of l_p (i.e. $l_p(rr',a) = l_p(r,a) + l_p(r',a)$) it immediately follows that this is true for *all* divisors r of N, not only prime ones. In particular, it is true for $r = N$ and since we can take $l_p(N,a) = 1$ we have $\chi(N) = \eta(\chi)$ and the first part of the proposition is proved.

For the second part, if ψ is of order p^{k_1} modulo p^{k_2} then if we take $\ell = l_p(r, \max(k_1,k_2))$ it is clear that $\psi\left(r^{p-1}\right) = \psi\left(N^{p-1}\right)^{\ell}$ and since $p-1$ is coprime to the order of ψ we immediately get the second part of the proposition. Note that we have implicitly used Lemma 9.1.10 in the proof of both parts. □

From this result, we obtain the following theorem which is very close to our final goal of proving N to be prime.

Theorem 9.1.12. *Let t be an even integer, let*

$$e(t) = 2 \prod_{\substack{q \text{ prime} \\ (q-1)\mid t}} q^{v_q(t)+1}$$

and assume that $(N, te(t)) = 1$. For each pair of prime numbers (p,q) such that $(q-1) \mid t$ and $p^k \| (q-1)$, let $\chi_{p,q}$ be a character modulo q of order p^k (for example $\chi_{p,q}\left(g_q^a\right) = \zeta_{p^k}^a$ if g_q is a primitive root modulo q). Assume that

(1) *For each pair (p,q) as above the character $\chi = \chi_{p,q}$ satisfies condition $(*_\beta)$ for some $\beta \notin \mathfrak{p}$ (but of course depending on p and q).*
(2) *For all primes $p \mid t$, condition $\mathcal{L}_p$ is satisfied.*

Then for every divisor r of N there exists an integer i such that $0 \leq i < t$ satisfying

$$r \equiv N^i \pmod{e(t)}.$$

Proof. From Proposition 9.1.11 and Lemma 9.1.10, there exists a sufficiently large a such that $\chi(r) = \chi(N)^{l_p(r,a)}$ for every a and every $\chi = \chi_{p,q}$. By the Chinese remainder Theorem 1.3.9, we can find $l(r)$ defined modulo t such that $l(r) \equiv l_p(r,a) \pmod{p^{v_p(t)}}$ for all primes p dividing t, hence since $p^k \mid (q-1) \mid t$, for all p and q as above we have

$$\chi_{p,q}(r) = \chi_{p,q}\left(N^{l(r)}\right).$$

Now I claim that $\chi_q = \prod_{p|(q-1)} \chi_{p,q}$ is a character of order exactly $q-1$. Indeed, if χ_0 is the trivial character modulo q, then $\chi_q^a = \chi_0$ implies that for every $p^k \| (q-1)$,

$$\chi_{p,q}^{a(q-1)/p^k} = \chi_0,$$

hence since $\chi_{p,q}$ is of order a power of p, hence prime to $(q-1)/p^k$, that $\chi_{p,q}^a = \chi_0$. This shows that $p^k \mid a$ since $\chi_{p,q}$ is of order exactly equal to p^k. Since this is true for every $p \mid q-1$, we have $(q-1) \mid a$, thus proving our assertion.

Hence, χ_q is a generator of the group of characters modulo q, and this implies that for any character χ_1 modulo q we have $\chi_1(r) = \chi_1\left(N^{l(r)}\right)$.

Now let χ be a character modulo $q^{v_q(t)+1+\delta}$ where $\delta = 0$ if $q > 2$, $\delta = 1$ if $q = 2$. We can write $\chi = \chi_1\chi_2$, where χ_1 is a character modulo q and χ_2 modulo $q^{v_q(t)+1+\delta}$ of order dividing $q^{v_q(t)+1+\delta-(1+\delta)} = q^{v_q(t)}$ (this follows from Theorem 1.4.1). Hence, if $q \nmid t$, $\chi = \chi_1$ so $\chi(r) = \chi(N^{l(r)})$. On the other hand, if $q \mid t$, then by assumption, condition $\mathcal{L}_q$ is satisfied. Hence, by Proposition 9.1.11 (2) we have

$$\chi_2(r) = \chi_2(N)^{l(r)} = \chi_2\left(N^{l(r)}\right)$$

since χ_2 is of order $q^{v_q(t)}$ and $l(r) \equiv l_q(r,a) \pmod{q^{v_q(t)}}$ for a sufficiently large. Therefore for every χ modulo $e(t)$ this equality is true, and this proves that

$$r \equiv N^{l(r)} \pmod{e(t)}.$$

Finally note that for every prime q such that $(q-1) \mid t$ we have

$$N^{(q-1)q^{v_q(t)}} \equiv 1 \pmod{q^{v_q(t)+1+\delta}}.$$

Hence, $N^t \equiv 1 \pmod{e(t)}$, so we may reduce the exponent $l(r)$ modulo t, thus proving the theorem. □

Corollary 9.1.13. *We keep all the notations and assumptions of the theorem. Set $r_i = N^i \bmod e(t)$, so that $0 < r_i < e(t)$. If $e(t) > \sqrt{N}$ and if for every i such that $0 < i < t$ we have $r_i = 1$ or $r_i = N$ or $r_i \nmid N$, then N is prime.*

Proof. If N was not prime, there would exist a prime divisor r of N such that $1 < r \leq \sqrt{N} < e(t)$, and by the theorem there would exist $i < t$ such that $r \equiv N^i \pmod{e(t)}$ hence $r = r_i$, contradiction. □

9.1.4 Checking Condition $\mathcal{L}_p$

We must now see how to check condition $\mathcal{L}_p$, and incidentally prove Lemma 9.1.10. We have the following result:

Lemma 9.1.14.

(1) *If $p \geq 3$, condition $\mathcal{L}_p$ is equivalent to the inequality*

$$v_p\left(r^{p-1}-1\right) \geq v_p\left(N^{p-1}-1\right).$$

(2) *For $p = 2$, condition $\mathcal{L}_2$ is equivalent to the inequality*

$$\max(v_2(r-1), v_2(r-N)) \geq v_2\left(N^2-1\right).$$

Proof. That condition $\mathcal{L}_p$ implies the above inequalities is trivial and left to the reader. Conversely, assume they are satisfied, and consider first the case $p \geq 3$. Set $u = v_p\left(N^{p-1}-1\right)$. Then it is easy to prove by induction on $a \geq 0$ that there exist integers x_i for $0 \leq i < l$ satisfying $0 \leq x_i < p$ and such that if we set $l_p(r, a+u) = \sum_{0 \leq i < l} x_i p^i$, we will have

$$r^{p-1} \equiv N^{(p-1)l_p(r,a+u)} \pmod{p^{a+u}}.$$

A similar induction works for $p = 2$ with $u = v_2\left(N^2-1\right)$ and $a+u$ replaced by $a+u-1$. This proves both the above lemma and Lemma 9.1.10 since the x_i are independent of a. □

Corollary 9.1.15. *If $p \geq 3$ and $N^{p-1} \not\equiv 1 \pmod{p^2}$, then condition $\mathcal{L}_p$ is satisfied.*

This is clear, since in this case $v_p\left(N^{p-1}-1\right) = 1$. □

This result is already useful for testing $\mathcal{L}_p$, but it is not a systematic way of doing so. Before giving a more systematic result, we need another lemma.

Lemma 9.1.16. *Let a and b be positive integers, and let x be in $\mathbb{Z}[\zeta_{p^k}, \zeta_q]$. Assume that for an integer r coprime to p we have the congruences*

$$x^a \equiv \eta_a \pmod{r} \qquad \text{and} \qquad x^b \equiv \eta_b \pmod{r},$$

where η_a and η_b are primitive roots of unity of order p^{l_a} and p^{l_b} respectively, where l_a and l_b are less than or equal to k.

Assume, in addition, that $l_a \geq l_b$ and $l_a \geq 1$. Then:

$$\begin{aligned} v_p(b) - v_p(a) &= l_a - l_b \qquad \text{if} \quad l_b > 0, \\ v_p(b) - v_p(a) &\geq l_a \qquad\quad \text{if} \quad l_b = 0. \end{aligned}$$

Proof. Write $a = p^{v_p(a)}m$, $b = p^{v_p(b)}n$ so $p \nmid mn$. If we had $v_p(a) > v_p(b)$, then, computing x^{an} in two different ways ($an = p^{v_p(a)-v_p(b)}bm$) we would obtain

$$\eta_a^n = \eta_b^{mp^{v_p(a)-v_p(b)}}$$

so $l_a < l_b$, contrary to our assumption. Hence, $v_p(b) \geq v_p(a)$, and we can now similarly compute x^{mb} in two different ways, giving

$$\eta_b^m = \eta_a^{np^{v_p(b)-v_p(a)}}.$$

This immediately implies the lemma. Note that a congruence between roots of unity of order a power of p is in fact an equality since p is coprime to r. □

The main result which allows us to test condition $\mathcal{L}_p$ is the following:

Proposition 9.1.17. *Assume that we can find a character χ modulo q, of order p^k and a $\beta \notin \mathfrak{p}$, for which $(*_\beta)$ is satisfied with $\eta(\chi)$ a* primitive *p^k-th root of unity. Then, if one of the following supplementary conditions is true, condition $\mathcal{L}_p$ is satisfied:*

(1) *If $p \geq 3$;*
(2) *If $p = 2$, $k = 1$ and $N \equiv 1 \pmod 4$;*
(3) *If $p = 2$, $k \geq 2$ and $q^{(N-1)/2} \equiv -1 \pmod N$.*

Proof. Assume that $p \geq 3$. By Lemma 9.1.8, if r is a prime divisor of N and if we set $x = \tau(\chi)^\beta$, then we have

$$x^{N^{(p-1)p^{k-1}}-1} \equiv \eta(\chi)^{\beta p^{k-1}} \pmod r$$

and

$$x^{r^{(p-1)p^{k-1}}-1} \equiv \chi(r)^{\beta p^{k-1}} \pmod r.$$

Since $\beta \notin \mathfrak{p}$, $\eta(\chi)^{\beta p^{k-1}}$ is a primitive p-th root of unity. From Lemma 9.1.16, we deduce that

$$v_p\Big(r^{(p-1)p^{k-1}} - 1\Big) - v_p\Big(N^{(p-1)p^{k-1}} - 1\Big) \geq 0.$$

But, since $p \geq 3$ for any integer m we have

$$v_p\left(m^{(p-1)p^{k-1}}-1\right)=k-1+v_p\left(m^{p-1}-1\right),$$

hence

$$v_p\left(r^{p-1}-1\right)\geq v_p\left(N^{p-1}-1\right)$$

and this proves the theorem in this case by Lemma 9.1.14.

The proof of the two other cases is similar and left to the reader (see Exercise 5). □

It is easy to show that if N is prime, one can always find a χ satisfying the hypotheses of Proposition 9.1.17. In practice, such a χ, if not already found among the χ which are used to test $(*_\beta)$, will be found after a few trials at most. Strictly speaking, however, this part of the algorithm makes it probabilistic, but in a weak sense. A non-probabilistic, but less practical version also exists (see [APR]).

9.1.5 The Use of Jacobi Sums

It is clear that we now have an asymptotically fast primality testing algorithm. In this form, however, it is far from being practical. The main reason is as follows: we essentially have to test a number of conditions of the form $(*_\beta)$ for certain β's and characters. This number is not that large, for example if N has less than 100 decimal digits, less than 80 tests will usually be necessary. The main problem lies in the computation of $\tau(\chi)^{\beta(N-\sigma_N)} \bmod N$. One needs to work in the ring $\mathbb{Z}[\zeta_{p^k},\zeta_q]$, and this will be hopelessly slow (to take again the case of $N<10^{100}$, we can take $t=5040$, hence p^k will be very small, more precisely $p^k\leq 16$, but q will be much larger, the largest value being $q=2521$). We must therefore find a better way to test these conditions. The reader may have wondered why we have carried along the element $\beta\in\mathbb{Z}[G]$, which up to now was not necessary. Now, however we are going to make a specific choice for β, and it will not be $\beta=1$. We have the following proposition.

Proposition 9.1.18. *Let χ be a character modulo q of order p^k, and let a and b be integers such that $p\nmid ab(a+b)$. Denote by E be the set of integers x such that $1\leq x<p^k$ and $p\nmid x$. Finally, let*

$$\alpha=\sum_{x\in E}\left\lfloor\frac{Nx}{p^k}\right\rfloor\sigma_x^{-1}$$

and

$$\beta=-\sum_{x\in E}\left(\left\lfloor\frac{xa}{p^k}\right\rfloor+\left\lfloor\frac{xb}{p^k}\right\rfloor-\left\lfloor\frac{x(a+b)}{p^k}\right\rfloor\right)\sigma_x^{-1}\ .$$

Then, we have

$$\tau(\chi)^{\beta(N-\sigma_N)}=j(\chi^a,\chi^b)^\alpha\ .$$

Proof. Set

$$\Theta = \sum_{x \in E} x\sigma_x^{-1} \in \mathbb{Z}[G].$$

An easy computation shows that for any integer r not divisible by p we have

$$\Theta(\sigma_r - r) = -p^k \sum_{x \in E} \left\lfloor \frac{rx}{p^k} \right\rfloor \sigma_x^{-1}.$$

Using this formula for $r = N$, a, b and $a + b$ (which are all coprime to p) we obtain

$$\Theta(N - \sigma_N) = p^k \alpha$$

and

$$\Theta(\sigma_a + \sigma_b - \sigma_{a+b}) = \Theta(\sigma_a - a + \sigma_b - b - (\sigma_{a+b} - (a+b))) = p^k \beta,$$

hence

$$\beta(N - \sigma_N) = \alpha(\sigma_a + \sigma_b - \sigma_{a+b}).$$

Now it follows from Proposition 9.1.6 that

$$j(\chi^a, \chi^b) = \tau(\chi)^{\sigma_a + \sigma_b - \sigma_{a+b}},$$

and our proposition follows. □

One sees from this proposition that if we can find suitable values of a and b, we can replace taking powers of $\tau(\chi)$, which are in a large ring, by powers of a Jacobi sum, which are in the much smaller ring $\mathbb{Z}[\zeta_{p^k}]$. This is the basic observation needed to make this test practical.

However this is not enough. First, note that the condition $p \nmid ab(a+b)$ excludes immediately the case $p = 2$, which will, as usual, have to be treated separately. Hence, we first assume that $p \geq 3$. Recall that to get anything useful from $(*_\beta)$ we must have $\beta \notin \mathfrak{p}$. This is easily dealt with by the following lemma.

Lemma 9.1.19. *With the notations of the above proposition, a necessary and sufficient condition for $\beta \notin \mathfrak{p}$ is that*

$$a^p + b^p \not\equiv (a+b)^p \pmod{p^2}.$$

Proof. If we set

$$K = -\sum_{x \in E} \left(\left\lfloor \frac{xa}{p^k} \right\rfloor + \left\lfloor \frac{xb}{p^k} \right\rfloor - \left\lfloor \frac{x(a+b)}{p^k} \right\rfloor \right) x^{-1}$$

where x^{-1} is an inverse of x modulo p^k, it is clear from the definition of $\mathfrak{p}$ that $\beta \notin \mathfrak{p}$ is equivalent to $p \nmid K$. Now by computing the product of ax for $x \in E$ in two different ways, it is easy to show that if $p \nmid a$

$$\sum_{x \in E} \left\lfloor \frac{xa}{p^k} \right\rfloor x^{-1} \equiv a \frac{a^{(p-1)p^{k-1}} - 1}{p^k} \pmod{p^k} \tag{A}$$

(see Exercise 1). The lemma follows immediately from this identity and the congruence

$$\frac{a^{(p-1)p^{k-1}} - 1}{p^k} \equiv \frac{a^{p-1} - 1}{p} \pmod{p}$$

(see Exercise 2). □

From this we obtain the following.

Proposition 9.1.20. *If* $3 \leq p < 6 \cdot 10^9$ *and* $p \neq 1093, 3511$, *we can take* $a = b = 1$. *In other words, if we take*

$$\beta = \sum_{p^k/2 < x < p^k, p \nmid x} \sigma_x^{-1}$$

then $\beta \notin \mathfrak{p}$ *and condition* $(*_\beta)$ *is equivalent to the congruence*

$$j(\chi, \chi)^\alpha \equiv \eta(\chi)^{-cN} \pmod{N},$$

where as before

$$\alpha = \sum_{x \in E} \left\lfloor \frac{Nx}{p^k} \right\rfloor \sigma_x^{-1}$$

and

$$c = 2 \frac{2^{(p-1)p^{k-1}} - 1}{p^k}.$$

Proof. By the preceding lemma, we can take $a = b = 1$ if we have $2^p \not\equiv 2 \pmod{p^2}$. This congruence is exactly the Wieferich congruence which occurs for the first case of Fermat's last theorem and has been tested extensively (see [Leh2]). One knows that the only solutions for $p < 6 \cdot 10^9$ are $p = 1093$ and $p = 3511$. The proposition now follows from Proposition 9.1.18 and formula (A) for $a = 2$. □

Note that the restriction on p in the above proposition is completely irrelevant in practice. Even if we were capable one day of using this test to prove the primality of numbers having 10^9 decimal digits, we would never need primes as large as 1093. This means that we have solved the practical problem of testing $(*_\beta)$ for $p \geq 3$.

The case $p = 2$ is a little more complicated, since we cannot use the above method. Let us first assume that $k \geq 3$. We must now consider the *triple Jacobi sum* defined by

$$j_3(\chi_1, \chi_2, \chi_3) = \sum_{x+y+z=1} \chi_1(x)\chi_2(y)\chi_3(z),$$

where the variables x, y and z range over $\mathbb{F}_q$. A similar proof to the proof of Proposition 9.1.6 shows that if $\chi_1\chi_2\chi_3$ is not the trivial character, then

$$j_3(\chi_1, \chi_2, \chi_3) = \frac{\tau(\chi_1)\tau(\chi_2)\tau(\chi_3)}{\tau(\chi_1\chi_2\chi_3)}$$

and in particular,

$$j_3(\chi, \chi, \chi) = \tau(\chi)^{3-\sigma_3}.$$

Now what we want is an analog of Proposition 9.1.18. This can be easily obtained for one half of the values of N as follows.

Proposition 9.1.21. *Let χ be a character modulo q of order 2^k with $k \geq 3$. Denote by E be the set of integers x such that $1 \leq x < 2^k$ and x congruent to 1 or 3 modulo 8. Finally, let*

$$\alpha = \sum_{x \in E} \left\lfloor \frac{Nx}{2^k} \right\rfloor \sigma_x^{-1}$$

and

$$\beta = \sum_{x \in E} \left\lfloor \frac{3x}{2^k} \right\rfloor \sigma_x^{-1}.$$

Then, if N is congruent to 1 or 3 modulo 8, we have

$$\tau(\chi)^{\beta(N-\sigma_N)} = j_3(\chi, \chi, \chi)^{\alpha}.$$

Furthermore, $\beta \notin \mathfrak{p}$.

Proof. The proof is essentially the same as that of Proposition 9.1.18, using $\Theta = \sum_{x \in E} x\sigma_x^{-1}$. The condition on N is necessary since $\Theta(\sigma_r - r)$ does not take any special form if r is not congruent to 1 or 3 modulo 8. The restriction to these congruences classes is also mandatory since $(\mathbb{Z}/2^k\mathbb{Z})^*$ is not cyclic but has cyclic subgroups of index 2. (We could also have taken for E those x congruent to 1 or 5 modulo 8, but that would have required the use of quintuple Jacobi sums). □

When N is congruent to 5 or 7 modulo 8, we use the following trick: $-N$ will be congruent to 1 or 3 modulo 8, hence $\Theta(\sigma_{-N} + N)$ will have a nice

form. But on the other hand, it is immediate to transform condition $(*_\beta)$ into a condition involving $\sigma_{-N} + N$:

$$\tau(\chi)^{\sigma_{-N}+N} = \tau(\chi)^{N-\sigma_N}\tau(\chi^N)\tau(\chi^{-N})$$

and by Proposition 9.1.6 we have

$$\tau(\chi^N)\tau(\chi^{-N}) = \chi(-1)q = -q \ ,$$

the last equality coming from $\chi(-1) = (-1)^{(q-1)/2^k} = -1$. This enables us to give a proposition analogous to Proposition 9.1.21 for N congruent to 5 or 7 modulo 8.

Proposition 9.1.22. *Let χ be a character modulo q of order 2^k with $k \geq 3$. Denote by E be the set of integers x such that $1 \leq x < 2^k$ and x congruent to 1 or 3 modulo 8. Finally, let*

$$\alpha_1 = \sum_{x \in E} \left(\left\lfloor \frac{Nx}{2^k} \right\rfloor + 1 \right) \sigma_x^{-1}$$

and

$$\beta = \sum_{x \in E} \left\lfloor \frac{3x}{2^k} \right\rfloor \sigma_x^{-1} \ .$$

Then, if N is congruent to 5 or 7 modulo 8, we have

$$\tau(\chi)^{\beta(N-\sigma_N)} = j_3(\chi,\chi,\chi)^{\alpha_1}(-q)^{-\beta} \ .$$

Furthermore, $\beta \notin \mathfrak{p}$.

The proof of this proposition follows immediately from what we have said before and is left to the reader. □

Corollary 9.1.23. *Let χ and E be as in the proposition. Set $\delta_N = 0$ if N is congruent to 1 or 3 modulo 8, $\delta_N = 1$ if N is congruent to 5 or 7 modulo 8. We may replace condition $(*_\beta)$ by the following condition:*

$$\left(j(\chi,\chi)j(\chi,\chi^2)\right)^{\alpha} j^{2\delta_n}\left(\chi^{2^{k-3}}, \chi^{3\cdot 2^{k-3}}\right) \equiv (-1)^{\delta_N}\eta(\chi)^{-cN} \pmod{N} \ ,$$

where

$$\alpha = \sum_{x \in E} \left\lfloor \frac{xN}{2^k} \right\rfloor \sigma_x^{-1}$$

and

$$c = 3\frac{3^{2^{k-2}} - 1}{2^k} \ .$$

Proof. Note first that using the formulas linking triple Jacobi sums with Gauss sums, and the analogous formula for ordinary Jacobi sums (Proposition 9.1.6), we have

$$j_3(\chi,\chi,\chi) = j(\chi,\chi)j(\chi,\chi^2)$$

and this is the most efficient way to compute j_3.

Now if N is congruent to 1 or 3 modulo 8, the result follows immediately from Proposition 9.1.21 and formula (A) for $a = 3$.

Assume now that N is congruent to 5 or 7 modulo 8. From Proposition 9.1.22, formula (A) and the identity

$$\sum_{x\in E}\left\lfloor\frac{3x}{2^k}\right\rfloor = 2^{k-2}-1,$$

we obtain

$$j_3(\chi,\chi,\chi)^{\alpha_1} \equiv \eta(\chi)^{-cN}(-q)^d$$

with $d = 2^{k-2}-1$. It is clear that the corollary will follow from this formula and the following lemma:

Lemma 9.1.24. *Set $\gamma = \sum_{x\in E}\sigma_x^{-1}$ and $d = 2^{k-2}-1$. We have the identity:*

$$j_3(\chi,\chi,\chi)^{\gamma} = q^d j^2\left(\chi^{2^{k-3}},\chi^{3\cdot 2^{k-3}}\right).$$

Proof. Using the formula expressing triple Jacobi sums in terms of Gauss sums, we have

$$j_3(\chi,\chi,\chi)^{\gamma} = \prod_{x\in E}\tau^2(\chi^x).$$

Now we have the following theorem, due to Hasse and Davenport (see for example [Was] and [Ire-Ros]).

Theorem 9.1.25 (Hasse-Davenport). *Let ψ be any character and χ_1 a character of order exactly equal to m. We have the identity*

$$\prod_{0\le x<m}\tau(\psi\chi_1^x) = -\tau(\psi^m)\,\psi^{-m}(m)\prod_{0\le x<m}\tau(\chi_1^x).$$

Applying this identity to $\psi = \chi^a$, $\chi_1 = \chi^{2^{k-l}}$, one easily shows by induction on l that

$$\prod_{0\le n<2^l}\tau^2\left(\chi^{a+n2^{k-l}}\right) = q^{2^l-1}\tau^2\left(\chi^{2^l a}\right)\chi(2)^{-al2^{l+1}}.$$

If we now take $l = k - 3$ and multiply the identities for $a = 1$ and $a = 3$, we easily obtain the lemma by using Proposition 9.1.6, thus proving our corollary. □

Note that one can give a direct proof of Lemma 9.1.24 without explicitly using the Hasse-Davenport theorem (see Exercise 3).

We have assumed that $k \geq 3$. What remains is the easy case of $k \leq 2$. Here we have the following proposition, whose proof is an immediate consequence of Proposition 9.1.6.

Proposition 9.1.26. *For $p = 2$ and $k = 1$, condition $(*_1)$ is equivalent to the congruence*

$$(-q)^{(N-1)/2} \equiv \eta(\chi) \pmod{N} .$$

For $p = 2$ and $k = 2$, condition $(*_1)$ is equivalent to the congruence

$$j(\chi,\chi)^{(N-1)/2} q^{(N-1)/4} \equiv \eta(\chi)^{-1} \pmod{N}$$

if $N \equiv 1 \pmod 4$, and to the congruence

$$j(\chi,\chi)^{(N+1)/2} q^{(N-3)/4} \equiv -\eta(\chi) \pmod{N}$$

if $N \equiv 3 \pmod 4$.

This ends our transformation of condition $(*_\beta)$ into conditions involving only the ring $\mathbb{Z}[\zeta_{p^k}]$.

9.1.6 Detailed Description of the Algorithm

We can now give a detailed and complete description of the Jacobi sum primality test.

Algorithm 9.1.27 (Precomputations). Let B be an upper bound on the numbers that we want to test for primality using the Jacobi sum test. This algorithm makes a number of necessary precomputations which do not depend on N but only on B.

1. [Find t] Using a table of $e(t)$, find a t such that $e^2(t) > B$.
2. [Compute Jacobi sums] For every prime q dividing $e(t)$ with $q \geq 3$, do as follows.

(1) Using Algorithm 1.4.4, compute a primitive root g_q modulo q, and a table of the function $f(x)$ defined for $1 \leq x \leq q-2$ by $1 - g_q^x = g_q^{f(x)}$ and $1 \leq f(x) \leq q-2$.

(2) For every prime p dividing $q-1$, let $k = v_p(q-1)$ and let $\chi_{p,q}$ be the character defined by $\chi_{p,q}\left(g_q^x\right) = \zeta_{p^k}^x$.

(3) If $p \geq 3$ or $p = 2$ and $k = 2$, compute

$$J(p,q) = j(\chi_{p,q}, \chi_{p,q}) = \sum_{1 \leq x \leq q-2} \zeta_{p^k}^{x+f(x)}.$$

If $p = 2$ and $k \geq 3$, compute $J(2,q)$ as above,

$$j(\chi_{2,q}^2, \chi_{2,q}) = \sum_{1 \leq x \leq q-2} \zeta_{2^k}^{2x+f(x)},$$

$$J_3(q) = j_3(\chi_{2,q}, \chi_{2,q}, \chi_{2,q}) = J(2,q) j(\chi_{2,q}^2, \chi_{2,q})$$

and

$$J_2(q) = j^2\left(\chi_{2,q}^{2^{k-3}}, \chi_{2,q}^{3\cdot 2^{k-3}}\right) = \left(\sum_{1 \leq x \leq q-2} \zeta_8^{3x+f(x)}\right)^2.$$

Note that it is very easy to build once and for all a table of $e(t)$. For example, $e(5040) \approx 1.532 \cdot 10^{52}$ hence $t = 5040$ can be used for numbers having up to 104 decimal digits, $e(720720) \approx 2.599 \cdot 10^{237}$, for numbers having up to 474 decimal digits (see however the remarks at the end of this section).

The Jacobi sum primality testing algorithm is then as follows.

Algorithm 9.1.28 (Jacobi Sum Primality Test). Let N be a positive integer. We assume that N is a strong pseudo-prime in 20 randomly chosen bases (so that N is almost certainly prime). We also assume that $N \leq B$ and that the precomputations described in the preceding algorithm have been made. This algorithm decides (rigorously!) whether N is prime or not.

1. [Check GCD] If $(te(t), N) > 1$, then N is composite and terminate the algorithm.
2. [Initialize] For every prime $p \mid t$, set $l_p \leftarrow 1$ if $p \geq 3$ and $N^{p-1} \not\equiv 1 \pmod{p^2}$, $l_p \leftarrow 0$ otherwise.
3. [Loop on characters] For each pair (p,q) of primes such that $p^k \| (q-1) \mid t$, execute step 4a if $p \geq 3$, step 4b if $p = 2$ and $k \geq 3$, step 4c if $p = 2$ and $k = 2$, step 4d if $p = 2$ and $k = 1$. Then go to step 5.

4a. [Check $(*_\beta)$ for $p \geq 3$] Let E be the set of integers between 0 and p^k which are not divisible by p. Set $\Theta \leftarrow \sum_{x \in E} x\sigma_x^{-1}$, $r \leftarrow N \bmod p^k$, $\alpha \leftarrow \sum_{x \in E} \left\lfloor \frac{rx}{p^k} \right\rfloor \sigma_x^{-1}$, and compute $s_1 \leftarrow J(p,q)^\Theta \bmod N$, $s_2 \leftarrow s_1^{\lfloor N/p^k \rfloor} \bmod N$, and finally $S(p,q) = s_2 J(p,q)^\alpha \bmod N$.

If there does not exist a p^k-th root of unity η such that $S(p,q) \equiv \eta \pmod{N}$, then N is composite and terminate the algorithm. If η exists and if it is a primitive p^k-th root of unity, set $l_p \leftarrow 1$.

4b.[Check $(*_\beta)$ for $p = 2$ and $k \geq 3$] Let E be the set of integers between 0 and 2^k which are congruent to 1 or 3 modulo 8. Set $\Theta \leftarrow \sum_{x \in E} x\sigma_x^{-1}$, $r \leftarrow N \bmod 2^k$, $\alpha \leftarrow \sum_{x \in E} \left\lfloor \frac{rx}{2^k} \right\rfloor \sigma_x^{-1}$, and compute $s_1 \leftarrow J_3(q)^\Theta \bmod N$, $s_2 \leftarrow s_1^{\lfloor N/p^k \rfloor} \bmod N$, and finally $S(2,q) = s_2 J_3(q)^\alpha J_2(q)^{\delta_N}$, where $\delta_N = 0$ if $r \in E$ (i.e. if N if congruent to 1 or 3 modulo 8), $\delta_N = 1$ otherwise.

If there does not exist a 2^k-th root of unity η such that $S(2,q) \equiv \eta \pmod N$, then N is composite and terminate the algorithm. If η exists and is a primitive 2^k-th root of unity, and if in addition $q^{(N-1)/2} \equiv -1 \pmod N$, set $l_2 \leftarrow 1$.

4c.[Check $(*_\beta)$ for $p = 2$ and $k = 2$] Set $s_1 \leftarrow J(2,q)^2 \cdot q \bmod N$, $s_2 \leftarrow s_1^{\lfloor N/4 \rfloor} \bmod N$, and finally $S(2,q) \leftarrow s_2$ if $N \equiv 1 \pmod 4$, $S(2,q) \leftarrow s_2 J(2,q)^2$ if $N \equiv 3 \pmod 4$.

If there does not exist a fourth root of unity η such that $S(2,q) \equiv \eta \pmod N$, then N is composite and terminate the algorithm. If η exists and is a primitive fourth root of unity (i.e. $\eta = \pm i$), and if in addition $q^{(N-1)/2} \equiv -1 \pmod N$, set $l_2 \leftarrow 1$.

4d.[Check $(*_\beta)$ for $p = 2$ and $k = 1$] Compute $S(2,q) \leftarrow (-q)^{(N-1)/2} \bmod N$. If $S(2,q) \not\equiv \pm 1 \pmod N$, then N is composite and terminate the algorithm. If $S(2,q) \equiv -1 \pmod N$ and $N \equiv 1 \pmod 4$, set $l_2 \leftarrow 1$.

5. [Check conditions $\mathcal{L}_p$] For every $p \mid t$ such that $l_p = 0$, do as follows. Choose random primes q such that $q \nmid e(t)$, $q \equiv 1 \pmod p$, $(q, N) = 1$, execute step 4a, 4b, 4c, 4d according to the value of the pair (p, q). To do this, we will have to compute a number of new Jacobi sums, since these will not have been precomputed, and we do this as explained in the precomputation algorithm.

If after a reasonable number of attempts, some l_p is still equal to 0, then output a message saying that the test has failed (this is highly improbable).

6. For $i = 1, \ldots, t-1$, compute (by induction of course, not by the binary powering algorithm) $r_i \leftarrow N^i \bmod e(t)$. If for some i, r_i is a non-trivial divisor of N, then N is composite and terminate the algorithm. Otherwise (i.e. if for every i either $r_i \nmid N$ or $r_i = 1$ or $r_i = N$), output the message that N is prime and terminate the algorithm.

9.1.7 Discussion

The above algorithm works already quite well both in theory and in practice. Pomerance and Odlyzko have shown that the running time of the Jacobi sum algorithm is

$$O((\ln N)^{C \ln\ln\ln N})$$

for some constant C. Hence this is almost (but not quite) a polynomial time algorithm. Many improvements are however still possible.

For example, it is not difficult to combine the Jacobi sum test with the information gained from the Pocklington $N-1$ and $N+1$ tests (Proposition

8.3.1). One can go even further and combine the test with the so-called Galois theory test. This has been done by Bosma and van der Hulst (see [Bos-Hul]).

Note also that the part of the algorithm which is the most time-critical is the computation of $s_2 \leftarrow s_1^{\lfloor N/p^k \rfloor}$. To do this, we of course use the fastest powering algorithms possible, in practice the 2^k-left to right Algorithm 1.2.4. But we must also do multiplications in the rings $\mathbb{Z}[\zeta_{p^k}]$ which is of dimension $n = \phi(p^k) = (p-1)p^{k-1}$ over $\mathbb{Z}$. A priori such a multiplication would require n^2 multiplications in $\mathbb{Z}$. Using the same tricks as explained in Section 3.1.2, it is possible to substantially decrease the number of necessary multiplications. Furthermore, special squaring routines must also be written. All this is explained in complete detail in [Coh-Len2] and [Coh-Len3].

Another important improvement uses an algorithm due to H. W. Lenstra (see [Len2]) for finding in polynomial time factors of N which are in a given residue class modulo s when $s > N^{1/3}$. This can be applied here, and allows us to replace the condition $e^2(t) > B$ of the precomputations by $e^3(t) > B$. This gives a substantial saving in time since one can choose a much smaller value of t. We give the algorithm here, and refer to [Len2] for its proof.

Algorithm 9.1.29 (Divisors in Residue Classes). Let r, s, N be integers such that $0 \leq r < s < N$, $(r,s) = 1$ and $s > \sqrt[3]{N}$. This algorithm determines all the divisors d of N such that $d \equiv r \pmod{s}$.

1. [Initialization] Using Euclid's extended Algorithm 1.3.6 compute u and v such that $ur + vs = 1$. Set $r' \leftarrow uN \bmod s$ (hence $0 \leq r' < s$), $a_0 \leftarrow s$, $b_0 \leftarrow 0$, $c_0 \leftarrow 0$, $a_1 \leftarrow ur' \bmod s$, $b_1 \leftarrow 1$, $c_1 \leftarrow u(N - rr')/s \bmod s$ and $j \leftarrow 1$. Finally, if $a_1 = 0$ set $a_1 = s$ (so $0 < a_1 \leq s$).
2. [Compute c] If j is even let $c \leftarrow c_j$. Otherwise, let $c \leftarrow c_j + s\lfloor (N + s^2(a_j b_j - c_j))/s^3 \rfloor$ and if $c < 2a_j b_j$ go to step 6.
3. [Solve quadratic equation] If $(cs + a_j r + b_j r')^2 - 4a_j b_j N$ is not the square of an integer, go to step 5. Otherwise, let t_1 and t_2 be the two (integral) solutions of the quadratic equation $T^2 - (cs + a_j r + b_j r')T + a_j b_j N = 0$.
4. [Divisor found?] If $a_j \mid t_1$, $b_j \mid t_2$, $t_1/a_j \equiv r \pmod{s}$ and $t_2/b_j \equiv r' \pmod{s}$, then output t_1/a_j as a divisor of N congruent to r modulo s.
5. [Other value of c] If j is even and $c > 0$, set $c \leftarrow c - s$ and go to step 3.
6. [Next j] If $a_j = 0$, terminate the algorithm. Otherwise, set $j \leftarrow j + 1$, and $q_j \leftarrow \lfloor a_{j-2}/a_{j-1} \rfloor$ if j is even, $q_j \leftarrow \lfloor (a_{j-2} - 1)/a_{j-1} \rfloor$ if j is odd. Finally, set $a_j \leftarrow a_{j-2} - q_j a_{j-1}$, $b_j \leftarrow b_{j-2} - q_j b_{j-1}$, $c_j \leftarrow c_{j-2} - q_j c_{j-1}$ and go to step 2.

Remarks.

(1) [Len2] also shows that under the conditions of this algorithm, there exist at most 11 divisors of N congruent to r modulo s.

(2) In step 4, t_2/b_j is a divisor of N congruent to r' modulo s. Since in the case of the Jacobi sum test $r = N^i \bmod s$ and so $r' = N^{1-i} \bmod s$, Lenstra's

algorithm allows us to test simultaneously two residue classes modulo s, reducing the time spent in step 6 of Algorithm 9.1.28.

9.2 The Elliptic Curve Test

We now come to the other modern primality test, based on the use of elliptic curves over finite fields. Here, instead of looking for suitably strong generalizations of Fermat's theorem in cyclotomic fields, or equivalently instead of implicitly using the multiplicative group of $\mathbb{F}_{N^d}$, we will use the group of points of elliptic curves over $\mathbb{F}_N$ itself.

Now recall that when we start using a primality test, we are already morally certain that our number N is prime, since it has passed the Rabin-Miller pseudo-primality test. Hence, we can work as if N was prime, for example by assuming that any non-zero element modulo N is invertible. In the unlikely event that some non-zero non-invertible element appears, we can immediately stop the algorithm since we know not only that N is composite, but even an explicit prime factor by taking a GCD with N.

We will consider an "elliptic curve over $\mathbb{Z}/N\mathbb{Z}$". What this means is that we consider a Weierstraß equation

$$y^2 = x^3 + ax + b\,, \qquad a, b \in \mathbb{Z}/N\mathbb{Z}\,, \quad (4a^3 + 27b^2) \in (\mathbb{Z}/N\mathbb{Z})^*\,.$$

(It is not necessary to consider a completely general Weierstraß equation since we may of course assume that $(N, 6) = 1$.)

We then add points on this curve *as if* N was prime. Since the group law involves only addition/subtraction/multiplication/division in $\mathbb{Z}/N\mathbb{Z}$, the only phenomenon which may happen if N is not prime is that some division is impossible, and in that case as already mentioned, we know that N is composite and we stop whatever algorithm we are executing.

Hence, from now on, we implicitly assume that all operations take place without any problems.

9.2.1 The Goldwasser-Kilian Test

The basic proposition which will enable us to prove that N is prime is the following analog of Pocklington's Theorem 8.3.1.

Proposition 9.2.1. *Let N be an integer coprime to 6 and different from 1. and E be an elliptic curve modulo N.*

Assume that we know an integer m and a point $P \in E(\mathbb{Z}/N\mathbb{Z})$ satisfying the following conditions.

(1) *There exists a prime divisor q of m such that*

$$q > (\sqrt[4]{N}+1)^2.$$

(2) $m \cdot P = O_E = (0:1:0)$.
(3) $(m/q) \cdot P = (x:y:t)$ *with* $t \in (\mathbb{Z}/N\mathbb{Z})^*$.

Then N is prime. (As above, it is assumed that all the computations are possible.)

Proof. Let p be a prime divisor of N. By reduction modulo p, we know that in the group $E(\mathbb{Z}/p\mathbb{Z})$, the image of P has order a divisor of m, but not a divisor of m/q since $t \in (\mathbb{Z}/N\mathbb{Z})^*$. Since q is a prime, this means that q divides the order of the image of P in $E(\mathbb{Z}/p\mathbb{Z})$, and in particular $q \leq |E(\mathbb{Z}/p\mathbb{Z})|$. By Hasse's Theorem 7.1.8, we thus have

$$q < (\sqrt{p}+1)^2.$$

Assume that N was not prime. We can then choose for p the smallest prime divisor of N which will be less than or equal to $\sqrt{N}$. Hence we obtain $q < (\sqrt[4]{N}+1)^2$, contradicting the hypothesis on the size of q and thus proving the proposition. □

For this proposition to be of any use, we must explain three things. First, how one chooses the elliptic curve, second how one finds P, and finally how one chooses m. Recall that for all these tasks, we may as well assume that N is prime, since this only helps us in making a choice. Only the above proposition will give us a *proof* that N is prime.

The only non-trivial choice is that of the integer m. First, we have:

Proposition 9.2.2. *Let N be a prime coprime to 6, E an elliptic curve modulo N and let*

$$m = |E(\mathbb{Z}/N\mathbb{Z})|.$$

If m has a prime divisor q satisfying

$$q > (\sqrt[4]{N}+1)^2,$$

then there exists a point $P \in E(\mathbb{Z}/N\mathbb{Z})$ such that

$$m \cdot P = O_E \quad \text{and} \quad (m/q) \cdot P = (x:y:t) \quad \text{with} \quad t \in (\mathbb{Z}/N\mathbb{Z})^*.$$

Proof. First note that any point P will satisfy $m \cdot P = O_E$. Second, since N is assumed here to be prime, $t \in (\mathbb{Z}/N\mathbb{Z})^*$ means $t \neq 0$ hence the second condition is $(m/q) \cdot P \neq O_E$.

Set $G = E(\mathbb{Z}/N\mathbb{Z})$ and assume by contradiction that for every $P \in G$ we have $(m/q) \cdot P = O_E$. This means that the order of any P is a divisor of m/q,

hence that the *exponent* of the Abelian group G divides m/q. (Recall that the exponent of an Abelian group is the LCM of the orders of the elements of the group.)

Now, by Theorem 7.1.9, we know that G is the product of at most two cyclic groups, i.e. that

$$G \simeq \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z} \quad \text{with} \quad d_2 \,|\, d_1$$

(and $d_2 = 1$ if G is cyclic). Hence the exponent of G is equal to d_1, while the cardinality of G is equal to $d_1 d_2 \leq d_1^2$. Thus we obtain

$$m = |G| \leq d_1^2 \leq (m/q)^2 ,$$

hence $q^2 \leq m$. Using our hypothesis on the size of q and Hasse's bound 7.1.8 on m, we obtain

$$(\sqrt[4]{N}+1)^2 < \sqrt{N}+1 ,$$

and this is clearly a contradiction, thus proving the proposition. □

We now know that Proposition 9.2.1 can in principle be applied to prove the primality of N, by choosing $m = |E(\mathbb{Z}/N\mathbb{Z})|$, where this cardinality is computed as if N was prime. But that is precisely the main question: how is this computed? We could of course use the baby-step giant-step Algorithm 7.4.12, but this is a $O(N^{1/4})$ algorithm, hence totally unsuitable.

The idea of Goldwasser and Kilian ([Gol-Kil]) is to make use of the remarkable algorithm of Schoof already mentioned in Section 7.4.3 ([Scho]), which computes $m = |E(\mathbb{Z}/N\mathbb{Z})|$ in time $O(\ln^8 N)$. Of course, this algorithm may fail since it is not absolutely certain that N is prime, but if it fails, we will know that N is composite.

Once m has been computed, we trial divide m by small primes, hoping that the unfactored part will be a large strong pseudo-prime. In fact, Goldwasser and Kilian's aim was purely theoretical, and in that case one looks for m equal to twice a strong pseudo-prime. If this is the case, and q is the large pseudo-prime that remains (large meaning larger than $(\sqrt[4]{N}+1)^2$ of course), we temporarily assume that q is prime, and look at random for a point P so as to satisfy the hypothesis of Proposition 9.2.1. This will be possible (and in fact quite easy) by Proposition 9.2.2.

If such a P is found, there remains the task of proving that our strong pseudo-prime q is prime. For this, we apply the algorithm recursively. Indeed, since $q \leq m/2 \leq (N+2\sqrt{N}+1)/2$, the size of N will decrease by a factor which is at least approximately equal to 2 at each iteration, hence the number of recursive uses of the algorithm will be $O(\ln N)$. We stop using this algorithm as soon as N becomes small enough so that other algorithms (even trial division!) may be used.

The algorithm may be formally stated as follows.

Algorithm 9.2.3 (Goldwasser-Kilian). Let N be a positive integer different from 1 and coprime to 6. This algorithm will try to prove that N is prime. If N is not a prime, the algorithm may detect it, or it may run indefinitely (hence we must absolutely use the Rabin-Miller test before entering this algorithm).

1. [Initialize] Set $i \leftarrow 0$ and $N_i \leftarrow N$.
2. [Is N_i small?] If $N_i < 2^{30}$, trial divide N_i by the primes up to 2^{15}. If N_i is not prime go to step 9.
3. [Choose a random curve] Choose a and b at random in $\mathbb{Z}/N_i\mathbb{Z}$, and check that $4a^3 + 27b^2 \in (\mathbb{Z}/N_i\mathbb{Z})^*$. Let E be the elliptic curve whose affine Weierstraß equation is $y^2 = x^3 + ax + b$.
4. [Use Schoof] Using Schoof's algorithm, compute $m \leftarrow |E(\mathbb{Z}/N_i\mathbb{Z})|$. If Schoof's algorithm fails go to step 9.
5. [Is m OK?] Check whether $m = 2q$ where q passes the Rabin-Miller test 8.2.2 (or more generally, trial divide m up to a small bound, and check that the remaining factor q passes the Rabin-Miller test and is larger than $(\sqrt[4]{N_i}+1)^2$). If this is not the case, go to step 3.
6. [Find P] Choose at random $x \in \mathbb{Z}/N_i\mathbb{Z}$ until the Legendre-Jacobi symbol $\left(\frac{x^3+ax+b}{N_i}\right)$ is equal to 0 or 1 (this will occur after a few trials at most). Then using Algorithm 1.5.1, compute $y \in \mathbb{Z}/N_i\mathbb{Z}$ such that $y^2 = x^3+ax+b$ (again, if this algorithm fails, go to step 9).
7. [Check P] Compute $P_1 \leftarrow m \cdot P$ and $P_2 \leftarrow (m/q) \cdot P$. If during the computations some division was impossible, go to step 9. Otherwise, check that $P_1 = O_E$, i.e. that $P_1 = (0 : 1 : 0)$ in projective coordinates. If $P_1 \neq O_E$, go to step 9. Finally, if $P_2 = O_E$, go to step 6.
8. [Recurse] Set $i \leftarrow i + 1$, $N_i \leftarrow q$ and go to step 2.
9. [Backtrack] (We are here when N_i is not prime, which is a very unlikely occurrence.) If $i = 0$, output a message saying that N is composite and terminate the algorithm. Otherwise, set $i \leftarrow i - 1$ and go to step 3.

Some remarks are in order. As stated in the algorithm, if N is not prime, the algorithm may run indefinitely and so should perhaps not be called an "algorithm" in our sense. Note however that it will never give a false answer. But even if N is prime, the algorithm is probabilistic in nature since we need to find an elliptic curve whose number of points has a special property, and in addition a certain point P on that curve. It can be shown that under reasonable hypotheses on the distribution of primes in short intervals, the expected running time of the algorithm is $O(\ln^{12} N)$, hence is polynomial in $\ln N$. Therefore it is asymptotically faster than the Jacobi sum test. Note however that the Goldwasser-Kilian test is not meant to be practical.

The sequence of primes $N_0 = N, N_1, \ldots N_i, \ldots$ together with the elliptic curves E_i, the points P_i and the cardinality m_i obtained in the algorithm is called a *primality certificate*. The reason for this is clear: although it may

have been difficult to find E_i, P_i or m_i, once they are given, to check that the conditions of Proposition 9.2.1 are satisfied (with $q = N_{i+1}$) is very easy, so anybody can prove to his or her satisfaction the primality of N using much less work than executing the algorithm. This is quite different from the Jacobi sum test where to check that the result given by the algorithm is correct, there is little that one can do but use a different implementation and run the algorithm again.

To finish this (sub)section, note that, as stated in the beginning of this chapter, an important theoretical advance has been made by Adleman and Huang.

Their idea is to use, in addition to elliptic curves, Jacobians of curves of genus 2, and a similar algorithm to the one above. Although their algorithm is also not practical, the important point is that they obtain a probabilistic primality testing algorithm which runs in polynomial time, in other words they prove Theorem 9.1. Note that the Goldwasser-Kilian test is not of this kind since only the expected running time is polynomial, but the worst case may not be.

9.2.2 Atkin's Test

Using the same basic idea, i.e. Proposition 9.2.1, Atkin has succeeded in finding a practical version of the elliptic curve test. It involves a number of new ideas. This version has been implemented by Atkin and by Morain, and has been able to prove the primality of *titanic numbers*, i.e. numbers having more than 1000 decimal digits. The Jacobi sum test could of course do the same, but time comparisons have not yet been done, although it seems that at least up to 800 digits the Jacobi sum test is slightly faster. Of course, since asymptotically Atkin's test is polynomial while the Jacobi sum test is not, the former must win for N sufficiently large.

The main (if not the sole) practical stumbling block in the algorithm of Goldwasser-Kilian is the computation of $m = |E(\mathbb{Z}/N\mathbb{Z})|$ using Schoof's algorithm. Although progress has been made in the direction of making Schoof's algorithm practical, for example by Atkin and Elkies, Atkin has found a much better idea.

Instead of taking random elliptic curves, we choose instead elliptic curves with complex multiplication by an order in a quadratic number field $K = \mathbb{Q}(\sqrt{D})$ where N splits as a product of two *elements*. This will enable us to use Theorem 7.2.15 which (if N is prime) gives us immediately the cardinality of $E(\mathbb{Z}/N\mathbb{Z})$.

The test proceeds as follows. As always we can work as if N was prime. We must first find a negative discriminant D such that N splits in the order of discriminant D as a product of two elements. This is achieved by using Cornacchia's Algorithm 1.5.3. Indeed, Cornacchia's algorithm gives us, if it exists, a solution to the equation $x^2 + dy^2 = 4p$, where $d = -D$, hence $\pi\overline{\pi} = p$, with

$$\pi = \frac{x + y\sqrt{D}}{2}.$$

Once such a D is found, using Theorem 7.2.15 we obtain that, if N is prime,

$$m = |E(\mathbb{Z}/N\mathbb{Z})| = N + 1 - \pi - \overline{\pi} = N + 1 - x$$

with the above notations, if E is an elliptic curve with complex multiplication by the order of discriminant D. We now check whether m satisfies the condition which will enable us to apply Proposition 9.2.1, i.e. that m is not prime, but its largest prime factor is larger than $(\sqrt[4]{N} + 1)^2$. Since we are describing a practical algorithm, this is done much more seriously than in Goldwasser-Kilian's test: we trial divide m up to a much higher bound, and then we can also use Pollard ρ and $p-1$ to factor m.

If m is not suitable, we still have at least another chance. Recall from Section 5.3 that we denote by $w(D)$ the number of roots of unity in the quadratic order of discriminant D, hence $w(D) = 2$ if $D < -4$, $w(-4) = 4$ and $w(-3) = 6$.

Then it can be shown that there exist exactly $w(D)$ isomorphism classes of elliptic curves modulo N with complex multiplication by the quadratic order of discriminant D. These correspond to the factorizations $N = (\zeta\pi)(\overline{\zeta\pi})$ where ζ runs over all $w(D)$-th roots of unity (in particular $\zeta = \pm 1$ if $D < -4$).

Hence we can compute $w(D)$ different values of m in this way and hope that at least one of them is suitable. If none are, we go on to another discriminant.

Therefore, let us assume that we have found a suitable value for m corresponding to a certain discriminant D. It remains to find explicitly the equations of elliptic curves modulo N with complex multiplication by the order of discriminant D.

Now since N splits in the order of discriminant D, we have $w(D) \mid N-1$ and there exist $(N-1)/2$ values of $g \in \mathbb{Z}/N\mathbb{Z}$ ($(N-1)/3$ if $D = -3$) such that $g^{(N-1)/p} \neq 1$ for each prime $p \mid w(D)$. Choose one of these values of g.

If $D = -4$ (resp. $D = -3$), then the four (resp. six) isomorphism classes of elliptic curves with complex multiplication by the order of discriminant -4 are given by the affine equations

$$y^2 = x^3 - g^k x \quad \text{for} \quad 0 \leq k \leq 3$$

(resp.

$$y^2 = x^3 - g^k \quad \text{for} \quad 0 \leq k \leq 5).$$

If D is not equal to -3 or -4, we set

$$c = j/(j - 1728), \quad \text{where} \quad j = j\left(\frac{D + \sqrt{D}}{2}\right)$$

is the j-invariant which corresponds to the order of discriminant D. Then the two isomorphism classes of elliptic curves with complex multiplication by the order of discriminant D can be given by the affine equations

$$y^2 = x^3 - 3cg^{2k}x + 2cg^{3k} \quad \text{for} \quad k = 0 \text{ or } 1.$$

Note that $j = j((D + \sqrt{D})/2)$ has been defined in Section 7.2.1 as a complex number, and not as an element of $\mathbb{Z}/N\mathbb{Z}$. Hence we must make sense of the above definition.

Recall that according to Theorem 7.2.14, j is an algebraic integer of degree exactly equal to $h(D)$. Furthermore, it can easily be shown that our hypothesis that N splits into a product of two elements is equivalent (if N is prime) to the fact that the minimal monic polynomial T of j in $\mathbb{Z}[X]$ splits completely modulo N as a product of linear factors. Since the roots of T in $\mathbb{C}$ are the conjugates of $j((D+\sqrt{D})/2)$, any one will define by the above equations the isomorphism classes of elliptic curves with complex multiplication by the order of discriminant D, hence we define j as being any of the $h(D)$ roots of $T(X)$ in $\mathbb{Z}/N\mathbb{Z}$.

Once the elliptic curve has been found, the rest of the algorithm proceeds as in the Goldwasser-Kilian algorithm, i.e. we must find a point P on the curve satisfying the required properties, etc ...

There are, however, two remarks to be made. First, we have $w(D)$ elliptic curves modulo N at our disposal, but a priori only one corresponds to a suitable value of m, and it is not clear which one. For $D = -3$ and $D = -4$, it is easy to give a recipe that will tell us which elliptic curve to choose. For $D < -4$, such a recipe is more difficult to find, and we then simply compute $m \cdot P$ for our suitable m and a random P on one of the two curves. If this is not equal to the identity, we are on the wrong curve. If it is equal to the identity, this does not prove that we are on the right curve, but if P has really been chosen randomly, we can probably still use the curve to satisfy the hypotheses of Proposition 9.2.1.

The second remark is much more important. To obtain the equation of the curve, it is necessary to obtain the value of j modulo N. This clearly is more difficult if the class number $h(D)$ is large. Hence, we start by considering discriminants whose class number is as small as possible. So we start by looking at the 13 quadratic orders with class number 1, then class number 2, etc ...

But now a new difficulty appears. The coefficients in the minimal polynomial T of j become large when the class number grows. Of course, they will afterwards be reduced modulo N, but to compute them we will need to use high precision computations of the values of $j(\tau)$ for every quadratic irrational τ corresponding to a reduced quadratic form of discriminant D. Since this computation is independent of N, it could be done only once and the results stored, but the coefficients are so large that even for a moderately sized table we would need an enormous amount of storage.

Several methods are available to avoid this. First, one can use the notion of *genus field* to reduce the computations to a combination of relative computations of smaller degree. Second, we can use *Weber functions*, which are meromorphic functions closely related to the function $j(\tau)$ and which have analogous arithmetic properties. In the best cases, these functions reduce the number of digits of the coefficients of the minimal polynomial T by a factor 24 (see Section 7.6.3).

All these tricks and many more, and the detailed implementation procedures, are described completely in [Atk-Mor] and in Morain's thesis [Mor2]. Here, we will simply give a formal presentation of Atkin's algorithm without any attempt at efficiency.

Algorithm 9.2.4 (Atkin). Given an integer N coprime to 6 and different from 1, this algorithm tries to prove that N is prime. It is assumed that N is already known to be a strong pseudo-prime in the sense of the Rabin-Miller test 8.2.2. We assume that we have a list of negative discriminants D_n $(n \geq 1)$ ordered by increasing computational complexity (for example as a first approximation by increasing class number).

1. [Initialize] Set $i \leftarrow 0$, $n \leftarrow 0$ and $N_i \leftarrow N$.

2. [Is N_i small?] If $N_i < 2^{30}$, trial divide N_i by the primes up to 2^{15}. If N_i is not prime go to step 14.

3. [Choose next discriminant] Let $n \leftarrow n+1$ and $D \leftarrow D_n$. If $\left(\frac{D}{N}\right) \neq 1$, go to step 3. Otherwise, use Cornacchia's Algorithm 1.5.3 to find a solution, if it exists, of the equation $x^2 + |D|y^2 = 4N$. If no solution exists, go to step 3.

4. [Factor m] For $m = N+1+x$, $m = N+1-x$ (and in addition for $m = N+1+2y$, $m = N+1-2y$ if $D = -4$, or $m = N+1+(x+3y)/2$, $m = N+1-(x+3y)/2$, $m = N+1+(x-3y)/2$, $m = N+1-(x-3y)/2$ if $D = -3$), factor m using trial division (up to 1000000, say), then Pollard ρ and $p-1$. It is worthwhile to spend *some* time factoring m here.

5. [Does a suitable m exist?] If, using the preceding step, for at least one value of m we can find a q dividing m which passes the Rabin-Miller test 8.2.2 and is larger than $(\sqrt[4]{N_i}+1)^2$, then go to step 6, otherwise go to step 3.

6. [Compute elliptic curve] If $D = -4$, set $a \leftarrow -1$ and $b \leftarrow 0$. If $D = -3$, set $a \leftarrow 0$, $b \leftarrow -1$. Otherwise, using Algorithm 7.6.1, compute the minimal polynomial $T \in \mathbb{Z}[X]$ of $j((D+\sqrt{D})/2)$. Then reduce T modulo N_i and let j be one of the roots of $\overline{T} = T \bmod N_i$ obtained by using Algorithm 1.6.1 (note that we know that $\overline{T} \mid X^{N_i} - X$ so the computation of $A(X)$ in step 1 of that algorithm is not necessary, we can simply set $A \leftarrow \overline{T}$). Then set $c \leftarrow j/(j-1728) \bmod N_i$, $a \leftarrow -3c \bmod N_i$, $b \leftarrow 2c \bmod N_i$.

7. [Find g] By making several random choices of g, find g such that g is a quadratic non-residue modulo N_i and in addition if $D = -3$, $g^{(N_i-1)/3} \not\equiv 1 \pmod{N_i}$.

8. [Find P] Choose at random $x \in \mathbb{Z}/N_i\mathbb{Z}$ until the Legendre-Jacobi symbol $\left(\frac{x^3+ax+b}{N_i}\right)$ is equal to 0 or 1 (this will occur after a few trials at most). Then using Algorithm 1.5.1, compute $y \in \mathbb{Z}/N_i\mathbb{Z}$ such that $y^2 = x^3 + ax + b$. (If this algorithm fails, go to step 14, but see also Exercise 6.) Finally, set $k \leftarrow 0$.
9. [Find right curve] Compute $P_2 \leftarrow (m/q) \cdot P$ and $P_1 \leftarrow q \cdot P_2$ on the curve whose affine equation is $y^2 = x^3 + ax + b$. If during the computations some division was impossible, go to step 14. If $P_1 = (0 : 1 : 0)$ go to step 12.
10. Set $k \leftarrow k + 1$. If $k \geq w(D)$ go to step 14, else if $D < -4$ set $a \leftarrow ag^2$, $b \leftarrow bg^3$, if $D = -4$ set $a \leftarrow ag$, if $D = -3$ set $b \leftarrow bg$ and go to step 8.
11. [Find a new P] Choose at random $x \in \mathbb{Z}/N_i\mathbb{Z}$ until the Legendre-Jacobi symbol $\left(\frac{x^3+ax+b}{N_i}\right)$ is equal to 0 or 1 (this will occur after a few trials at most). Then using Algorithm 1.5.1, compute $y \in \mathbb{Z}/N_i\mathbb{Z}$ such that $y^2 = x^3 + ax + b$ (if this algorithm fails, go to step 14). If $P_1 \neq (0 : 1 : 0)$ go to step 10.
12. [Check P] If $P_2 = O_E$, go to step 11.
13. [Recurse] Set $i \leftarrow i + 1$, $N_i \leftarrow q$ and go to step 2.
14. [Backtrack] (We are here when N_i is not prime, which is unlikely.) If $i = 0$, output a message saying that N is composite and terminate the algorithm. Otherwise, set $i \leftarrow i - 1$ and go to step 3.

Most remarks that we have made about the Goldwasser-Kilian algorithm are still valid here. In particular, this algorithm is probabilistic, but its expected running time is polynomial in $\ln N$. More important, it is practical, and as already mentioned, it has been used to prove the primality of numbers having more than 1000 decimal digits, by using weeks of workstation time.

Also, as for the Goldwasser-Kilian test, it gives a certificate of primality for the number N, hence the primality of N can be re-checked much faster.

9.3 Exercises for Chapter 9

1. a) Let p be a prime, E the set of integers x such that $1 \leq x < p^k$ and $p \nmid x$, and a an integer such that $p \nmid a$. By computing the product of ax for $x \in E$ in two different ways, show that we have

$$\sum_{x \in E} \left\lfloor \frac{xa}{p^k} \right\rfloor x^{-1} \equiv a \frac{a^{(p-1)p^{k-1}} - 1}{p^k} \pmod{p^k}.$$

b) Generalize this result, replacing p^k by an arbitrary integer m and the condition $p \nmid a$ by $(a, m) = 1$.

2. Show that if p is an odd prime and $p \nmid a$, we have

$$\frac{a^{(p-1)p^{k-1}} - 1}{p^k} \equiv \frac{a^{p-1} - 1}{p} \pmod{p}.$$

3. Prove Lemma 9.1.24 without explicitly using the Hasse-Davenport relations.

4. (Wolstenholme's theorem)

a) Let p be a prime, and set

$$\sum_{1\le x\le p-1} \frac{1}{x} = \frac{A_p}{B_p}$$

where A_p and B_p are coprime integers. By first adding together the terms for x and for $p-x$, show that $p^2 \mid A_p$ (note that $p \mid A_p$ is immediate).

b) As in Exercise 1, generalize to arbitrary integers m, replacing $\sum_{1\le x\le p-1}$ by $\sum_{1\le x\le m,(x,m)=1}$.

5. Let $a \in \mathbb{Z}$ and assume that $a^{(N-1)/2} \equiv -1 \pmod{N}$.

a) Show that for every $r \mid N$ we have $v_2(r-1) \ge v_2(N-1)$.

b) Show that equality holds if and only if $\left(\frac{a}{r}\right) = -1$, and in particular that $\left(\frac{a}{N}\right) = -1$.

c) If $N \equiv 1 \pmod 4$ show that condition $\mathcal{L}_2$ is satisfied.

d) If $N \equiv 3 \pmod 8$ and $a = 2$ show that condition $\mathcal{L}_2$ is satisfied.

6. Show how to avoid the search in step 8 of Algorithm 9.2.4 by setting $d \leftarrow x^3 + ax + b$ for some x and modifying the equation of the curve as in step 3 of Algorithm 7.4.12.

7. Let χ be a character modulo q, where q is not necessarily prime. We will say that χ is *primitive* if for all divisors d of q such that $d < q$, there exists an x such that $x \equiv 1 \pmod{d}$ and $\chi(x) \ne 0$ and 1. Set $\zeta = e^{2i\pi/q}$, and $\psi(a) = \sum_{x\in(\mathbb{Z}/q\mathbb{Z})^*} \chi(x)\zeta^{ax}$.

a) Let a be such that $d = (a,q) = 1$. Show that $\psi(a) = \overline{\chi}(a)\tau(\chi)$.

b) Assume that χ is a primitive character and that $d = (a,q) > 1$. Show that there exists a $u \in (\mathbb{Z}/q\mathbb{Z})^*$ such that $au = d$. Deduce from this that $\psi(a) = 0$, and hence that the formula $\psi(a) = \overline{\chi}(a)\tau(\chi)$ is still valid.

c) Show that if χ is a primitive character modulo q which is not necessarily a prime, we still have $|\tau(\chi)| = \sqrt{q}$.

8. Let χ be a primitive character modulo $q > 1$, as defined in the preceding exercise, and set $S(x) = \sum_{n\le x} \chi(n)$.

a) Using the preceding exercise, give an explicit formula for $\tau(\overline{\chi})S(x)$.

b) Deduce that

$$\sqrt{q}\,|S(x)| \le \sum_{1\le m<q,\ m\ne q/2} \frac{1}{\sin\frac{\pi m}{q}}.$$

c) Show finally the *Polya—Vinogradov inequality*

$$|S(x)| = \left|\sum_{1\le n\le x} \chi(n)\right| \le \sqrt{q}\log q.$$

Chapter 10

Modern Factoring Methods

The aim of this chapter is to give an overview of the fastest factoring methods known today. This could be the object of a book in itself, hence it is unreasonable to be as detailed here as we have been in the preceding chapters. In particular, most methods will not be written down as formal algorithms as we have done before. We hope however that we will have given sufficient information so that the reader may understand the methods and be able to implement them, at least in unoptimized form. The reader who wants to implement these methods in a more optimized form is urged to read the abundant literature after reading this chapter, before doing so.

10.1 The Continued Fraction Method

We will start this survey of modern factoring methods by the continued fraction factoring algorithm (CFRAC). Although superseded by better methods, it is important for two reasons. First, because it was historically the first algorithm which is asymptotically of sub-exponential running time (although this is only a heuristic estimate and was only realized later), and also because in the late 60's and 70's it was the main factoring method in use. The second reason is that it shares a number of properties with more recent factoring methods: it finds a large number of congruences modulo N, and the last step consists in Gaussian elimination over the field $\mathbb{Z}/2\mathbb{Z}$. Since the ideas underlying it are fairly simple, it is also a natural beginning.

The main idea of CFRAC, as well as the quadratic sieve algorithm (Section 10.4) or the number field sieve (Section 10.5), is to find integers x and y such that

$$x^2 \equiv y^2 \pmod{N}, \qquad x \not\equiv \pm y \pmod{N}.$$

Since $x^2 - y^2 = (x-y)(x+y)$, it is clear that the $\gcd(N, x+y)$ will be a non-trivial factor of N.

Now finding randomly such integers x and y is a hopeless task. The trick, common to the three factoring methods mentioned above, is to find instead congruences of the form

$$x_k^2 \equiv (-1)^{e_{0k}} p_1^{e_{1k}} p_2^{e_{2k}} \cdots p_m^{e_{mk}} \pmod{N}$$

where the p_i are "small" prime numbers. If we find sufficiently many such congruences, by Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$ we may hope to find a relation of the form

$$\sum_{1\le k\le n} \epsilon_k(e_{0k},\cdots,e_{mk}) \equiv (0,\cdots,0) \pmod 2$$

where $\epsilon_k = 0$ or 1, and then if

$$x = \prod_{1\le k\le n} x_k^{\epsilon_k}, \quad y = (-1)^{v_0} p_1^{v_1}\cdots p_m^{v_m}$$

where $\sum_k \epsilon_k(e_{0k},\cdots,e_{mk}) = 2(v_0,\cdots,v_m)$, it is clear that we have $x^2 \equiv y^2 \pmod N$. This splits N if, in addition $x \not\equiv \pm y \pmod N$, condition which will usually be satisfied.

The set of primes p_i (for $1 \le i \le m$) which are chosen to find the congruences is called the *factor base*. We will see in each of the factoring methods how to choose it in an optimal manner. These methods differ mainly in the way they generate the congruences.

The CFRAC method, stemming from ideas of Legendre, Kraitchik, Lehmer and Powers, and developed for computer use by Brillhart and Morrison ([Bri-Mor]), consists in trying to find *small* values of t such that $x^2 \equiv t \pmod N$ has a solution. In that case, since t is small, it has a reasonably good chance of being a product of the primes of our factor base, thus giving one of the sought for congruences.

Now if t is small and $x^2 \equiv t \pmod N$, we can write $x^2 = t + kd^2N$ for some k and d, hence $(x/d)^2 - kN = t/d^2$ will be small. In other words, the rational number x/d is a good approximation to the quadratic number $\sqrt{kN}$. Now it is well known (and easy, see [H-W]) that continued fraction expansions of real numbers give good (and in a certain sense the best) rational approximations. This is the basic idea behind CFRAC. We compute the continued fraction expansion of $\sqrt{kN}$ for a number of values of k. This gives us good rational approximations P/Q, say, and we then try to factor the corresponding integer $t = P^2 - Q^2kN$ (which will be not too large) on our factor base. If we succeed, we will have a new congruence.

Now from Section 5.7, we know that it is easy to compute the continued fraction expansion of a quadratic number, using no real approximations, but only rather simple integer arithmetic. Note that although we know that the expansion will be ultimately periodic (in fact periodic after one term in the case of $\sqrt{kN}$), this is completely irrelevant for us since, except for very special numbers, we will never compute the expansion on a whole period or even a half period. The main point which I stress again is that the expansion can be computed *simply*, in contrast with more general numbers.

The formulas of Sections 5.6 and 5.7, adapted to our situation, are as follows. Let $\tau = (-U + \sqrt{D})/2V$ be a quadratic number in the interval $[0,1[$ with $4V \mid U^2 - D$ and $V > 0$ (hence $|U| < \sqrt{D}$). We have

$$1/\tau = \frac{2V(U+\sqrt{D})}{D-U^2} = \frac{U+\sqrt{D}}{2V'}$$

where $V' = (D-U^2)/(4V)$ is a positive integer. Hence, if we set

$$a = \left\lfloor \frac{1}{\tau} \right\rfloor = \left\lfloor \frac{U+\sqrt{D}}{2V'} \right\rfloor,$$

then

$$\frac{-U+\sqrt{D}}{2V} = \frac{1}{a + \dfrac{-U'+\sqrt{D}}{2V'}} = \frac{1}{a+\tau'}$$

with $U' = U - 2aV'$. Clearly $\tau' \in [0,1[$, and since $4VV' = D - U^2 \equiv D - U'^2 \pmod{4V'}$ the conditions on (U,V) are also satisfied for (U',V') hence the process can continue. Thus we obtain the continued fraction expansion of our initial τ.

Note we have simply repeated the proof of Proposition 5.6.6 (2) that if a quadratic form $f = (V, U, (U^2-D)/(4V))$ is reduced, then $\rho(f)$ is also reduced. In addition, Proposition 5.6.3 tells us that we will always have U and V less than $\sqrt{D}$ if we start with a reduced form. This will be the case for the form corresponding to the quadratic number $\tau = \sqrt{D} - \lfloor\sqrt{D}\rfloor$. If we denote by a_n (resp U_n, V_n, τ_n), the different quantities a, U, V and τ occurring in the above process, we have, with the usual notation of continued fractions

$$\sqrt{D} = [a_0, a_1, a_2, \cdots, a_n + \tau_n]$$

where we have set $a_0 = \lfloor\sqrt{D}\rfloor$. Hence, if we set

$$[a_0, a_1, \cdots, a_n] = \frac{P_n}{Q_n},$$

we have the usual recursions

$$(P_{n+1}, Q_{n+1}) = a_{n+1}(P_n, Q_n) + (P_{n-1}, Q_{n-1})$$

with $(P_{-1}, Q_{-1}) = (1,0)$, $(P_0, Q_0) = (a_0, 1)$.

Returning to our factoring process, we apply this continued fraction algorithm to $D = kN$ for squarefree values of k such that $kN \equiv 0$ or $1 \pmod 4$. Then P_n/Q_n will be a good rational approximation to $\sqrt{kN}$, hence $t = P_n^2 - Q_n^2 kN$ will not be too large (more precisely $|t| < 2\sqrt{kN}$, see Proposition 5.7.3), and we can try to factor it on our factor base. For every success, we obtain a congruence

$$x^2 \equiv (-1)^{e_0} p_1^{e_1} p_2^{e_2} \cdots p_m^{e_m} \pmod N$$

as above, and as already explained, once we have obtained at least $m+2$ such congruences then by Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$ we can obtain a

congruence $x^2 \equiv y^2 \pmod{N}$, and hence (usually) a non-trivial splitting of N.

Remarks.

(1) For a prime p to be useful in our factor base we must have $\left(\frac{kN}{p}\right) = 0$ or 1. Indeed, if $p \mid P_n^2 - Q_n^2 kN$, we cannot have $p \mid Q_n$ otherwise P_n and Q_n would not be coprime. Hence kN is congruent to a square modulo p, which is equivalent to my claim.

(2) An important improvement to the method of factoring on a fixed factor base is to use the so-called *large prime variation* which is as follows. A large number of residues will not quite factor completely on our factor base, but will give congruences of the form $x^2 \equiv Fp \pmod{N}$ where F does factor completely and p is a large prime number not in the factor base. A single such relation is of course useless. But if we have two with the same large prime p, say $x_1^2 \equiv F_1 p \pmod{N}$ and $x_2^2 \equiv F_2 p \pmod{N}$, we will have $(x_1 x_2/p)^2 \equiv F_1 F_2 \pmod{N}$ which is a useful relation.

Now since p is large (typically more than 10^5), it could be expected that getting the same p twice is very rare. That this is not true is an instance of the well known "birthday paradox". What it says in our case is that if k numbers are picked at random among integers less than some bound B, then if $k > B^{1/2}$ (approximately) there will be a probability larger than $1/2$ that two of the numbers picked will be equal (see Exercise 5). Hence this large prime variation will give us quite a lot of extra relations essentially for free.

(3) Another important improvement to CFRAC is the so-called *early abort strategy*. It is based on the following idea. Most of the time is being spent in the factorization of the residues (this is why methods using sieves such as MPQS or NFS are so much faster). Instead of trying to factor completely on our factor base, we can decide that if after a number of primes have been tried the unfactored portion is too large, then we should abort the factoring procedure and generate the next residue. With a suitable choice of parameters, this gives a considerable improvement.

(4) Finally, note that the final Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$ is a non-trivial task since the matrices involved can be huge. These matrices are however very sparse, hence special techniques apply. See for example the "intelligent Gaussian elimination" method used by LaMacchia and Odlyzko ([LaM-Odl]), as well as [Cop1], [Cop2].

10.2 The Class Group Method

10.2.1 Sketch of the Method

The continued fraction method, as well as the more recent quadratic sieve (Section 10.4) or number field sieve (Section 10.5) have sub-exponential running time, which make them quite efficient, but require also sub-exponential space.

The class group method due to Schnorr and Lenstra was the first sub-exponential method which required a negligible amount of space, say polynomial space. The other prominent method having this characteristic is the elliptic curve method (see Section 10.3).

Note that we name this method after Schnorr and Lenstra since they published it ([Schn-Len]), but essentially the same method was independently discovered and implemented by Atkin and Rickert, who nicknamed it SPAR (Shanks, Pollard, Atkin, Rickert).

The idea of the method is as follows. We have seen in Section 8.6 that the determination of the 2-Sylow subgroup of the class group of the quadratic field $\mathbb{Q}(\sqrt{-N})$ is equivalent to knowing all the factorizations of N. In a manner analogous to the continued fraction method, we consider the class numbers $h(-kN)$ of $\mathbb{Q}(\sqrt{-kN})$ for several values of k. Then, if $h(-kN)$ is smooth, we will be able to apply the $p-1$ method, replacing the group $\mathbb{F}_p^*$ by the class group of $\mathbb{Q}(\sqrt{-kN})$. As for the $p-1$ method, this will enable us to compute the (unknown) order of a group, the only difference being that from the order of $\mathbb{F}_p^*$ we split N by computing a GCD with N, while in our case we will split N by using ambiguous forms.

Since we will use $p-1$-type methods, we need to specify the bounds B_1 (for the first stage), and B_2 (for the second stage). Since we have a large number of groups at our disposal, we will be able to create a method which will be a systematic factoring method by choosing B_1 and B_2 appropriately, since we can hope that $h(-kN)$ will be smooth for a value of k which is not too large.

To choose these values appropriately, we need a fundamental theorem about smooth numbers. The upper bound was first proved by de Bruijn ([de-Bru]), and the complete result by Canfield, Erdős and Pomerance ([CEP]). It is as follows.

Theorem 10.2.1 (Canfield, Erdős, Pomerance). *Let*

$$\psi(x,y) = |\{n \leq x, n \text{ is } y\text{-smooth }\}|.$$

Then if we set $u = \ln x / \ln y$, *we have*

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

uniformly for $x \to \infty$ *if* $(\ln x)^\epsilon < u < (\ln x)^{1-\epsilon}$ *for a fixed* $\epsilon \in (0,1)$.

In particular, if we set

$$L(x) = e^{\sqrt{\ln x \ln \ln x}},$$

then

$$\psi(x, L(x)^a) = xL(x)^{-1/(2a)+o(1)}.$$

Now heuristic methods (see Section 5.10 and [Coh-Len1]) seem to indicate that class numbers are not only as smooth, but even slightly smoother than average. Furthermore, it is not difficult to see that there is little quantitative difference between B-smoothness and B-powersmoothness. Hence, it is not unreasonable to apply Theorem 10.2.1 to estimate the behavior of powersmoothness of class numbers. In addition, the class number $h(-N)$ is $O(N^{1/2+\epsilon})$ (for example $h(-N) < \frac{1}{\pi}\sqrt{N}\ln N$, see Exercise 27 of Chapter 5).

Hence, if we take $x = \sqrt{N}$ and $B = L(x)^a$, we expect that the probability that a given class number of size around x is B-powersmooth should be at least $L(x)^{-1/(2a)+o(1)}$, hence the expected number of values of k which we will have to try before hitting a B-powersmooth number should be approximately $L(x)^{1/(2a)+o(1)}$. (Note that the class number $h(-kN)$ is still $O(N^{1/2+\epsilon})$ for such values of k.) Hence, ignoring step 2 of the $p-1$ algorithm (which in any case influences only on the O constant, not the exponents), the expected running time with this choice of B is $O(L(x)^{a+1/(2a)+o(1)})$, and this is minimal for $a = 1/\sqrt{2}$. Since $L(x)^{1/\sqrt{2}} \approx L(N)^{1/2}$, we see that the optimal choice of B is approximately $L(N)^{1/2}$, and the expected running time is $L(N)^{1+o(1)}$. Note also that the storage is negligible.

10.2.2 The Schnorr-Lenstra Factoring Method

We now give the algorithm. Note that contrary to the $p-1$ method, we do not need to do any backtracking since if x is an ambiguous form which is not the unit form (i.e. is of order exactly equal to 2), so is x^r for any odd number r).

Algorithm 10.2.2 (Schnorr-Lenstra). Let N be a composite number. This algorithm will attempt to split N. We assume that we have precomputed a table $p[1], \ldots, p[k]$ of all the primes up to $L(N)^{1/2}$.

1. [Initialize] Set $B \leftarrow \lfloor L(N)^{1/2} \rfloor$, $K \leftarrow 1$, $e \leftarrow \lfloor \lg B \rfloor$.
2. [Initialize for K] Let $D = -KN$ if $KN \equiv 3 \pmod 4$, $D = -4KN$ otherwise.
3. [Choose form] Let f_p be a random primeform of discriminant D (see Algorithm 5.4.10). Set $x \leftarrow f_p$, $c \leftarrow 0$ and $i \leftarrow 1$.
4. [Next prime] Set $i \leftarrow i+1$. If $i > k$, set $K \leftarrow K+1$ and go to step 2. Otherwise, set $q \leftarrow p[i]$, $q_1 \leftarrow q$, $l \leftarrow \lfloor B/q \rfloor$.

5. [Compute power] While $q_1 \leq l$, set $q_1 \leftarrow q \cdot q_1$. Then, set $x \leftarrow x^{q_1}$ (powering in the class group), $c \leftarrow c+1$ and if $c < 20$ go to step 4.
6. [Success?] Set $e_1 \leftarrow 0$, and while x is not an ambiguous form and $e_1 < e$ set $x \leftarrow x^2$ and $e_1 \leftarrow e_1 + 1$. Now if x is not an ambiguous form, set $c \leftarrow 0$, and go to step 4.
7. [Finished?] (Here x is an ambiguous form.) Find the factorization of KN corresponding to x. If this does not split N (for example if x is the unit form), go to step 3. Otherwise, output a non-trivial factor of N and terminate the algorithm.

Note that if in step 7 we obtain an ambiguous form which does not succeed in splitting N, this very probably still means that the K used is such that $h(-KN)$ is B-powersmooth. Therefore we must keep this value of K and try another random form in the group, but we should not change the group anymore. Note also that the first prime tried in step 4 is $p[2] = 3$, and *not* $p[1] = 2$.

To give a numerical example of the numbers involved, for $N = 10^{60}$, which is about the maximum size of numbers which one can factor in a reasonable amount of time with this method, we have $B \approx 178905$, and since we need the primes only up to B, this is quite reasonable. In fact, it is better to take a lower value of $B_1 = B$, and use the second stage of the $p-1$ method with quite a larger value for B_2. This reduces the expected running time of the algorithm, but the optimal values to take are implementation dependent. We leave as an exercise for the reader the incorporation of step 2 of the $p-1$ method into this algorithm, using these remarks (see Exercise 2).

As in all algorithms using class groups of quadratic fields, the basic operation in this algorithm is composition of quadratic forms. Even with the use of optimized methods like NUDUPL and NUCOMP (Algorithms 5.4.8 and 5.4.9), this is still a slow operation. Hence, although this method is quite attractive because of its running time, which is as good as all the other modern factoring algorithms with the exception of the number field sieve, and although it uses little storage, to the author's knowledge it has never been used intensively in factoring projects. Indeed, the elliptic curve method for instance has the same characteristics as the present one as far as speed and storage are concerned, but the group operations on elliptic curves can be done faster than in class groups, especially when (as will be the case), several curves have to be dealt with simultaneously (see Section 10.3).

Also note that it has been proved by Lenstra and Pomerance that for composite numbers of a special form the running time of this algorithm is very poor (i.e. exponential time).

10.3 The Elliptic Curve Method

10.3.1 Sketch of the Method

We now come to another method which also uses ideas from the $p-1$-method, but uses the group of points of an elliptic curve over $\mathbb{Z}/p\mathbb{Z}$ instead of the group $(\mathbb{Z}/p\mathbb{Z})^*$. This method, due to H. W. Lenstra, is one of the three main methods in use today, together with the quadratic sieve (see Section 10.4) and the number field sieve (see Section 10.5). In addition it possesses a number of properties which make it useful even if it is only used in conjunction with other algorithms. Like the class group method, it requires little storage and has a similar expected running time. Unique among modern factoring algorithms however, it is sensitive to the size of the prime divisors. In other words, its running time depends on the size of the smallest prime divisor p of N, and not on N itself. Hence, it can be profitably used to remove "small" factors, after having used trial division and the Pollard ρ method 8.5.2. Without too much trouble, it can find prime factors having 10 to 20 decimal digits. On the other hand, it very rarely finds prime factors having more than 30 decimal digits. This means that if N is equal to a product of two roughly equal prime numbers having no special properties, the elliptic curve method will not be able to factor N if it has more than, say, 70 decimal digits. In this case, one should use the quadratic sieve or the number field sieve.

We now describe the algorithm. As in the class group algorithm, for simplicity we give only the version which uses stage 1 of the $p-1$-method, the extension to stage 2 being straightforward.

Recall that the group law on an elliptic curve of the form $y^2 = x^3+ax+b$ is given by formulas which generically involve the expression $(y_2-y_1)/(x_2-x_1)$. This makes perfect sense in a field (when $x_2 \neq x_1$), but if we decide to work in $\mathbb{Z}/N\mathbb{Z}$, this will not always make sense since $x_2 - x_1$ will not always be invertible when $x_2 \neq x_1$. But this is exactly the point: if $x_2 - x_1$ is not invertible in $\mathbb{Z}/N\mathbb{Z}$ with $x_2 \neq x_1$, this means that $(x_2 - x_1, N)$ is a non-trivial divisor of N, and this is what we want. Hence we are going to work on an elliptic curve modulo N (whatever that is, we will define it in Section 10.3.2), and work as if N is prime. Everything will work out as long as every non-zero number modulo N that we encounter is invertible. As soon as it does not work out, we have found a non-trivial factorization of N. At this point, the reader may wonder what elliptic curves have to do with all this. We could just as well choose numbers x at random modulo N and compute (x, N), hoping to find a non-trivial divisor of N. It is easy to see that this would be a $O(N^{1/2+\epsilon})$ algorithm, totally unsuitable. But if N has a prime divisor p such that our elliptic curve E has a smooth number of points modulo p, the $p-1$-method will discover this fact, i.e. find a power of a point giving the unit element of the curve modulo p. This means that we will have some x_1 and x_2 such that $x_1 \equiv x_2 \pmod{p}$, hence $(x_2 - x_1, N) > 1$, and as with all these methods, this is in fact equal to a non-trivial divisor of N. This means it is reasonable to expect that something will break down, which is what we hope in this case.

Before turning to the detailed description of the algorithm, it is instructive to compare the different methods using the $p-1$-idea. For this discussion, we assume that we obtain exactly the prime p which is at the basis of the method. Let B be the stage 1 bound, $M = \mathrm{lcm}[1..B]$, and let G be the underlying group and a an element of G.

(1) In the $p-1$ method itself (or its variants like the $p+1$ method), $G = \mathbb{F}_p^*$ (or $G = \mathbb{F}_{p^2}^*$), and we obtain p directly as $\gcd(a^M - 1, N)$.
(2) In the class group method, $G = Cl(\mathbb{Q}(\sqrt{-KN}))$ for a suitable K, and we obtain p indirectly through the correspondence between a factorization $KN = p \cdot KN/p$ and some ambiguous forms x in G, which is obtained as $a^{M/2^t}$ for a suitable value of t.
(3) In the elliptic curve method, $G = E(\mathbb{F}_p)$ and we obtain p indirectly because of the impossibility of computing a^M modulo N (that is, we encountered a non-invertible element).

We see that the reasons why we obtain the factorization of N are quite diverse. The running time is essentially governed by the abundance of smooth numbers, i.e. by the theorem of Canfield, Erdős and Pomerance, and so it is not surprising that the running time of the elliptic curve method will be similar to that of the class group method, with the important difference of being sensitive to the size of p.

10.3.2 Elliptic Curves Modulo N

Before giving the details of the method, it is useful to give some idea of projective geometry over $\mathbb{Z}/N\mathbb{Z}$ when N is not a prime. When N is a prime, the projective line over $\mathbb{Z}/N\mathbb{Z}$ can simply be considered as the set $\mathbb{Z}/N\mathbb{Z}$ to which is added a single "point at infinity", hence has $N+1$ elements. When N is not a prime, the situation is more complicated.

Definition 10.3.1. *We define* projective n-space over $\mathbb{Z}/N\mathbb{Z}$ *as follows.*

Let $E = \{(x_0, x_1, \ldots, x_n) \in (\mathbb{Z}/N\mathbb{Z})^{n+1},\ \gcd(x_0, x_1, \ldots, x_n, N) = 1\}$. *If* $\mathcal{R}$ *is the relation on* E *defined by multiplication by an invertible element of* $\mathbb{Z}/N\mathbb{Z}$, *then* $\mathcal{R}$ *is an equivalence relation, and we define*

$$\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z}) = E/\mathcal{R},$$

i.e. the set of equivalence classes of E *modulo the relation* $\mathcal{R}$.

We will denote by $(x_0 : x_1 : \cdots : x_n)$ *the equivalence class in* $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ *of* $(x_0, x_1, \ldots, x_n)$.

Remarks.

(1) Note that even though the x_i are in $\mathbb{Z}/N\mathbb{Z}$, it makes sense to take their GCD together with N by taking any representatives in $\mathbb{Z}$ and then computing the GCD.

(2) We recover the usual definition of projective n-space over a field when N is prime.

(3) The set $(\mathbb{Z}/N\mathbb{Z})^n$ can be naturally embedded into $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ by sending $(x_0, x_1, \ldots, x_{n-1})$ to $(x_0 : x_1 : \cdots : x_{n-1} : 1)$. This subset of $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ will be called for our purposes *its* affine subspace, and denoted $\mathbb{P}_n^{\mathrm{Aff}}(\mathbb{Z}/N\mathbb{Z})$, although it is not canonically defined.

(4) If p is a prime divisor of N (or in fact any divisor), there exists a natural map from $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ to $\mathbb{P}_n(\mathbb{Z}/p\mathbb{Z})$ induced by reducing projective coordinates modulo p. Then P belongs to $\mathbb{P}_n^{\mathrm{Aff}}(\mathbb{Z}/N\mathbb{Z})$ if and only if the reduction of P modulo every prime divisor p of N belongs to $\mathbb{P}_n^{\mathrm{Aff}}(\mathbb{Z}/p\mathbb{Z})$.

(5) When N is a prime, we have a natural decomposition $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z}) = \mathbb{P}_n^{\mathrm{Aff}}(\mathbb{Z}/N\mathbb{Z}) \cup \mathbb{P}_{n-1}(\mathbb{Z}/N\mathbb{Z})$, by identifying $(x_0 : x_1 : \cdots : x_{n-1})$ with $(x_0 : x_1 : \cdots : x_{n-1} : 0)$. In the general case, this is no longer true. We can still make the above identification of $\mathbb{P}_{n-1}$ with a subspace of $\mathbb{P}_n$. (It is easy to check that it is compatible with the equivalence relation defining the projective spaces.) There is however a third subset which enters, made up of points $P = (x_0 : x_1 : \cdots : x_n)$ such that x_n is neither invertible nor equal to 0 modulo N, i.e. such that (x_n, N) is a non-trivial divisor of N. We will call this set the *special subset*, and denote it by $\mathbb{P}_n^s(\mathbb{Z}/N\mathbb{Z})$. For any subset E of $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ we will denote by E^{Aff}, E_{n-1} and E^s the intersection of E with $\mathbb{P}_n^{\mathrm{Aff}}$, $\mathbb{P}_{n-1}$ and $\mathbb{P}_n^s$ respectively. Hence, we have the disjoint union

$$E = E^{\mathrm{Aff}} \cup E_{n-1} \cup E^s.$$

Let us give an example. The projective line over $\mathbb{Z}/6\mathbb{Z}$ has 12 elements, which are $(0 : 1)$, $(1 : 1)$, $(2 : 1)$, $(3 : 1)$, $(4 : 1)$, $(5 : 1)$, $(1 : 2)$, $(3 : 2)$, $(5 : 2)$, $(1 : 3)$, $(2 : 3)$ and $(1 : 0)$ (denoting by the numbers 0 to 5 the elements of $\mathbb{Z}/6\mathbb{Z}$). The first 6 elements make up the affine subspace, and the last element $(1 : 0)$ corresponds to the usual point at infinity, i.e. to $\mathbb{P}_0$. The other 5 elements are the special points.

It is clear that finding an element in the special subset of $\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})$ will immediately factor N, hence the special points are the ones which are interesting for factoring.

We leave as an exercise for the reader to show that

$$|\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})| = N^n \prod_{p|N} \left(1 + \frac{1}{p} + \cdots + \frac{1}{p^n}\right),$$

and in particular

$$|\mathbb{P}_1(\mathbb{Z}/N\mathbb{Z})| = N \prod_{p|N} \left(1 + \frac{1}{p}\right)$$

(see Exercise 6).

Definition 10.3.2. *Let N be a positive integer coprime to 6. We define an elliptic curve E over $\mathbb{Z}/N\mathbb{Z}$ as a projective equation of the form*

$$y^2t = x^3 + axt^2 + bt^3$$

where $(x : y : t)$ *are the projective coordinates, and* a *and* b *are elements of* $\mathbb{Z}/N\mathbb{Z}$ *such that* $4a^3 + 27b^2$ *is invertible modulo* N.

As usual, by abuse of notation we shall use affine equations and affine coordinates even though it is understood that we work in the projective plane.

Now if N is a prime, the above definition is indeed the definition of an elliptic curve over the field $\mathbb{F}_N$. When N is not a prime the reduction maps modulo the prime divisors p of N clearly send $E(\mathbb{Z}/N\mathbb{Z})$ into $E(\mathbb{Z}/p\mathbb{Z})$. (Note that the condition that $4a^3 + 27b^2$ is invertible modulo N ensures that the reduced curves will all be elliptic curves.) Hence, as with any other set we can write

$$E(\mathbb{Z}/N\mathbb{Z}) = E^{\mathrm{Aff}} \cup E_1 \cup E^s \ ,$$

and E^s is the set of points $(x : y : t)$ such that t is neither invertible nor equal to 0 modulo N. This means, in particular, that the reduction of $(x : y : t)$ modulo p will not always be in the affine part modulo p.

Warning. Note that if the reduction of $(x : y : t)$ modulo every prime divisor p of N is the point at infinity, this does *not* imply that t is equal to 0 modulo N. What it means is that t is divisible by all the primes dividing N, and this implies $t \equiv 0 \pmod{N}$ only if N is squarefree.

Now we can use the addition laws given by Proposition 7.1.7 to try and define a group law on $E(\mathbb{Z}/N\mathbb{Z})$. They will of course not work as written, since even if $x_1 \neq x_2$, $x_1 - x_2$ may not be invertible modulo N. There are two ways around this. The first one, which we will not use, is to define the law on the projective coordinates. This can be done, and involves essentially looking at 9 different cases (see [Bos]). We then obtain a true group law, and on the affine part it is clear that the reduction maps modulo p are compatible with the group laws.

The second way is to stay ignorant of the existence of a complete group law. After all, we only want to factor N. Hence we use the formulas of Proposition 7.1.7 as written. If we start with two points in the affine part, their sum P will either be in the affine part, or of the form $(x : y : 0)$ (i.e. belong to E_1), or finally in the special part. If P is in the special part, we immediately split N since (t, N) is a non-trivial factor of N. If $P = (x : y : 0)$, then note that since $P \in E(\mathbb{Z}/N\mathbb{Z})$ we have $x^3 \equiv 0 \pmod{N}$. Then either $x \equiv 0 \pmod{N}$, corresponding to the non-special point at infinity of E, or (x, N) is a non-trivial divisor of N, and again we will have succeeded in splitting N.

10.3.3 The ECM Factoring Method of Lenstra

Before giving the algorithm in detail, we must still settle a few points. First, we must explain how to choose the elliptic curves, and how to choose the stage 1 bound B.

As for the choice of elliptic curves, one can simply choose $y^2 = x^3 + ax + 1$ which has the point $(0 : 1 : 1)$ on it, and a is small. For the stage 1 bound, since the number of points of E modulo p is around p by Hasse's theorem, one expects $E(\mathbb{Z}/p\mathbb{Z})$ to be $L(p)^a$-powersmooth with probability $L(p)^{-1/(2a)+o(1)}$ by the Canfield-Erdős-Pomerance theorem, hence if we take $B = L(p)^a$ we expect to try $L(p)^{1/(2a)+o(1)}$ curves before getting a smooth order, giving as total amount of work $L(p)^{a+1/(2a)+o(1)}$ group operations on the curve. This is minimal for $a = 1/\sqrt{2}$, giving a running time of $L(p)^{\sqrt{2}+o(1)}$ group operations.

Since, when N is composite, there exists a $p \mid N$ with $p \leq \sqrt{N}$, this gives the announced running time of $L(N)^{1+o(1)}$. But of course what is especially interesting is that the running time depends on the size of the smallest prime factor of N, hence the ECM can be used in a manner similar to trial division. In particular, contrary to the class group method, the choice of B should be done not with respect to the size of N, but, as in the original $p-1$ method, with respect to the amount of time that one is willing to spend, more precisely to the approximate size of the prime p one is willing to look for.

For example, if we want to limit our search to primes less than 10^{20}, one can take $B = 12000$ since this is close to the value of $L(10^{20})^{1/\sqrt{2}}$, and we expect to search through 12000 curves before successfully splitting N. Of course, in actual practice the numbers will be slightly different since we will also use stage 2. The algorithm is then as follows.

Algorithm 10.3.3 (Lenstra's ECM). Let N be a composite integer coprime to 6, and B be a bound chosen as explained above. This algorithm will attempt to split N. We assume that we have precomputed a table, $p[1], \ldots, p[k]$ of all the primes up to B.

1. [Initialize curves] Set $a \leftarrow 0$ and let E be the curve $y^2t = x^3 + axt^2 + t^3$.
2. [Initialize] Set $x \leftarrow (0 : 1 : 1)$, $i \leftarrow 0$.
3. [Next prime] Set $i \leftarrow i+1$. If $i > k$, set $a \leftarrow a+1$ and go to step 2. Otherwise, set $q \leftarrow p[i]$, $q_1 \leftarrow q$, $l \leftarrow \lfloor B/q \rfloor$.
4. [Compute power] While $q_1 \leq l$, set $q_1 \leftarrow q \cdot q_1$. Then, try to compute $x \leftarrow q_1 \cdot x$ (on the curve E) using the law given by Proposition 7.1.7. If the computation never lands in the set of special points or the $n-1$ part of E (i.e. if one does not hit a non-invertible element t modulo N), go to step 3.
5. [Finished?] (Here the computation has failed, which is what we want.) Let t be the non-invertible element. Set $g \leftarrow (t, N)$ (which will not be equal to 1). If $g < N$, output g and terminate the algorithm. Otherwise, set $a \leftarrow a+1$ and go to step 2.

Note that when $g = N$ in step 5, this means that our curve has a smooth order modulo p, hence, as with the class group algorithm, we should keep the same curve and try another point. Finding another point may however not be easy since N is not prime, so there is no easy way to compute a square root modulo N (this is in fact essentially equivalent to factoring N, see Exercise

1). Therefore we have no other choice but to try again. As usual, this is an exceedingly rare occurrence, and so in practice it does not matter.

10.3.4 Practical Considerations

The ECM algorithm as given above in particular involves one division modulo N per operation on the elliptic curve, and this needs approximately the same time as computing a GCD with N. Thus we are in a similar situation to the Schnorr-Lenstra Algorithm 10.2.2 where the underlying group is a class group and the group operation is composition of quadratic forms, which also involves computing one, and sometimes two GCD's. Hence, outside from the property that ECM usually gives small factors faster, it seems that the practical running time should be slowed down for the same reason, i.e. the relative slowness of the group operation.

In the case of the ECM method however, many improvements are possible which do not apply to the class group method. The main point to notice is that here all the GCD's (or extended GCD's) are with the *same* number N. Hence, we can try grouping all these extended GCD's by working with several curves in parallel. That this can easily be done was first noticed by P. Montgomery. We describe his trick as an algorithm.

Algorithm 10.3.4 (Parallel Inverse Modulo N). Given a positive integer N and k integers $a_1, \ldots, a_k$ which are not divisible by N, this algorithm either outputs a non-trivial factor of N or outputs the inverses $b_1, \ldots, b_k$ of the a_i modulo N.

1. [Initialize] Set $c_1 \leftarrow a_1$ and for $i = 2, \ldots, k$ set $c_i \leftarrow c_{i-1} \cdot a_i \bmod N$.
2. [Apply Euclid] Using one of Euclid's extended algorithms of Section 1.3, compute (u, v, d) such that $uc_k + vN = d$ and $d = (c_k, N)$. If $d = 1$ go to step 3. Otherwise, if $d = N$, then set $d \leftarrow (a_i, N)$ for $i = 1, \ldots, k$ until $d > 1$ (this will happen). Output d as a non-trivial factor of N and terminate the algorithm.
3. [Compute inverses] For $i = k, k-1, \ldots i = 2$ do the following. Output $b_i \leftarrow uc_{i-1} \bmod N$, and set $u \leftarrow ua_i \bmod N$. Finally, output $b_1 \leftarrow u$ and terminate the algorithm.

Proof. We clearly have $c_i = a_1 \cdots a_i \bmod N$, hence at the beginning of step 3 we have $u = (a_1 \cdots a_i)^{-1} \bmod N$, showing that the algorithm is valid. □

Let us see the improvements that this algorithm brings. The naïve method would have required k extended Euclid to do the job. The present algorithm needs only 1 extended Euclid, plus $3k - 3$ multiplications modulo N. Hence, it is superior as soon as 1 extended Euclid is slower than 3 multiplications modulo N, and this is almost always the case.

Now recall from Chapter 7 that the computation of the sum of two points on an elliptic curve $y^2 = x^3 + ax + b$ requires the computation of $m = (y_2 -$

$y_1)(x_2 - x_1)^{-1}$ if the points are distinct, $m = (3x_1^2 + a)(2y_1)^{-1}$ if the points coincide, plus 2 multiplications modulo N and a few additions or subtractions. Since the addition/subtraction times are small compared to multiplication modulo N, we see that by using Montgomery's trick on a large number C of curves, the actual time taken for a group operation on the curve in the context of the ECM method is $6+T/C$ multiplications modulo N when the points are distinct, or $7 + T/C$ when they are equal, where T is the ratio between the time of an extended GCD with N and the time of a multiplication modulo N. (Incidentally, note that in every other semi-group that we have encountered, including $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{Z}[X]$ or even class groups, squaring is always faster than general multiplication. In the case of elliptic curves, it is the opposite.) If we take C large enough (say $C = 50$) this gives numbers which are not much larger than 6 (resp. 7), and this is quite reasonable.

Another way to speed up group computations on elliptic curves modulo N is to use projective coordinates instead of affine ones. The big advantage is then that no divisions modulo N are required at all. Unfortunately, since we must now keep track of three coordinates instead of two, the total number of operations increases, and the best that one can do is 12 multiplications modulo N when the points are distinct, 13 when they are equal (see Exercise 3). Thanks to Montgomery's trick, this is worse than the affine method when we work on many curves simultaneously.

By using other parametrizations of elliptic curves than the Weierstraß model $y^2 = x^3 + ax + b$, one can reduce the number 12 to 9 (see [Chu] and Exercise 4), but this still does not beat the $6 + T/C$ above when C is large. Hence, in practice I suggest using affine coordinates on the Weierstraß equation and Montgomery's trick.

Finally, as for the class group method, it is necessary to include a stage 2 into the algorithm, as for the $p - 1$ method. The details are left to the reader (see [Mon2], [Bre2]).

As a final remark in this section, we note that one can try to use other algebraic groups than elliptic curves, for example Abelian varieties. D. and G. Chudnovsky have explored this (see [Chu]), but since the group law requires a lot more operations modulo N, this does not seem to be useful in practice.

10.4 The Multiple Polynomial Quadratic Sieve

We now describe the quadratic sieve factoring algorithm which, together with the elliptic curve method, is the most powerful general factoring method in use at this time (1994). (The number field sieve has been successfully applied to numbers of a special form, the most famous being the ninth Fermat number $2^{2^9} + 1 = 2^{512} + 1$, a 155 digit number, but for general numbers, the quadratic sieve is still more powerful in the feasible range.) This method is due to C. Pomerance, although some of the ideas were already in Kraitchik.

10.4.1 The Basic Quadratic Sieve Algorithm

As in the continued fraction method CFRAC explained in Section 10.1, we look for many congruences of the type

$$x_k^2 \equiv (-1)^{e_{0k}} p_1^{e_{1k}} p_2^{e_{2k}} \cdots p_m^{e_{mk}} \pmod{N}$$

where the p_i are "small" prime numbers, and if we have enough, a Gaussian stage will give us a non-trivial congruence $x^2 \equiv y^2 \pmod{N}$ and hence a factorization of N. The big difference with CFRAC is the way in which the congruences are generated. In CFRAC, we tried to keep $x^2 \bmod N$ as small as possible so that it would have the greatest possible chance of factoring on our factor base of p_i. We of course assume that N is not divisible by any element of the factor base.

Here we still want the $x^2 \bmod N$ to be not too large but we allow residues larger than $\sqrt{N}$ (although still $O(N^{1/2+\epsilon})$. The simplest way to do this is to consider the polynomial

$$Q(a) = \left(\left\lfloor \sqrt{N} \right\rfloor + a\right)^2 - N.$$

It is clear that $Q(a) \equiv x^2 \pmod{N}$ for $x = \left\lfloor \sqrt{N} \right\rfloor + a$ and as long as $a = O(N^{\epsilon})$, we will have $Q(a) = O(N^{1/2+\epsilon})$.

Although this is a simpler and more general way to generate small squares modulo N than CFRAC, it is not yet that interesting. The crucial point, from which part of the name of the method derives, is that contrary to CFRAC we do not need to (painfully) factor all these $x^2 \bmod N$ over the factor base. (In fact, most of them do not factor so this would represent a waste of time.) Here, since $Q(a)$ is a polynomial with integer coefficients, we can use a *sieve*. Let us see how this works. Assume that for some number m we know that $m \mid Q(a)$. Then, for every integer k, $m \mid Q(a + km)$ automatically. To find an a (if it exists) such that $m \mid Q(a)$ is of course very easy since we solve $x^2 \equiv N \pmod{m}$ using the algorithm of Exercise 30 of Chapter 1, and take $a = x - \left\lfloor \sqrt{N} \right\rfloor \bmod m$.

Since we are going to sieve, without loss of generality we can restrict to sieving with prime powers $m = p^k$. If p is an odd prime, then $x^2 \equiv N \pmod{p^k}$ has a solution (in fact two) if and only if $\left(\frac{N}{p}\right) = 1$, so we include only those primes in our factor base (this was also the case in the CFRAC algorithm) and we compute explicitly the two possible values of $a \pmod{p^k}$ such that $p^k \mid Q(a)$, say a_{p^k} and b_{p^k}. If $p = 2$ and $k \geq 3$, then $x^2 \equiv N \pmod{2^k}$ has a solution (in fact four) if and only if $N \equiv 1 \pmod 8$ and we again compute them explicitly. Finally, if $p = 2$ and $k = 2$, we take $x = 1$ if $N \equiv 1 \pmod 4$ (otherwise a does not exist) and if $p = 2$ and $k = 1$ we take $x = 1$.

Now for a in a very long interval (the sieving interval), we compute very crudely $\ln |Q(a)|$. (As we will see, an absolute error of 1 for instance is enough,

hence we certainly will *not* use the internal floating point log but some ad hoc program.) We then store this in an array indexed by a. For every prime p in our factor base, and more generally for small prime powers when p is small (a good rule of thumb is to keep all possible p^k less than a certain bound), we subtract a crude approximation to $\ln p$ to every element of the array which is congruent to a_{p^k} or to b_{p^k} modulo p^k (this is the sieving part). When all the primes of the factor base have been removed in this way, it is clear that a $Q(a)$ will factor on our factor base if and only if what remains at index a in our array is close to 0 (if the logs were exact, it would be exactly zero). In fact, if $Q(a)$ does not factor completely, then the corresponding array element will be at least equal to $\ln B$ (where B is the least prime which we have not included in our factor base), and since this is much larger than 1 this explains why we can take very crude approximations to logs.

It can be shown on heuristic grounds, again using the theorem of Canfield, Erdős and Pomerance, that using suitable sieving intervals and factor bases, the running time is of the form $O(L(N)^{1+o(1)})$. Although this is comparable to the class group or ECM methods, note that the basic operation in the quadratic sieve is a single precision subtraction, and it is difficult to have a faster basic operation than that! As a consequence, for practical ranges (say up to 100 decimal digits) the quadratic sieve runs faster than the other methods that we have seen, although as already explained, ECM may be lucky if N has a relatively small prime divisor.

The method that we have just briefly explained is the basic quadratic sieve (QS). Many improvements are possible. The two remarks made at the end of Section 10.1 also apply here. First, only primes p such that $p = 2$ or $\left(\frac{N}{p}\right) = 1$ need to be taken in the prime base (or more generally $\left(\frac{kN}{p}\right) = 0$ or 1 if a multiplier is used). Second, the large prime variation is just as useful here as before. (This is also the case for the number field sieve, and more generally for any algorithm which uses in some way factor bases, for example McCurley or Buchmann's sub-exponential algorithms for class group and regulator computation.)

10.4.2 The Multiple Polynomial Quadratic Sieve

There is however a specific improvement to the quadratic sieve which explains the first two words of the complete name of the method (MPQS). The polynomial $Q(a)$ introduced above is nice, but unfortunately it stands all alone, hence the values of $Q(a)$ increase faster than we would like. The idea of the Multiple Polynomial Quadratic Sieve is to use several polynomials Q so that the size of $Q(a)$ can be kept as small as possible. The following idea is due to P. Montgomery.

We will take quadratic polynomials of the form $Q(x) = Ax^2 + 2Bx + C$ with $A > 0$, $B^2 - AC > 0$ and such that $N \mid B^2 - AC$. This gives congruences just as nicely as before since

$$AQ(x) = (Ax + B)^2 - (B^2 - AC) \equiv (Ax + B)^2 \pmod{N}.$$

In addition, we want the values of $Q(x)$ to be as small as possible on the sieving interval. If we want to sieve on an interval of length $2M$, it is therefore natural to center the interval at the minimum of the function Q, i.e. to sieve in the interval

$$I = [-B/A - M, -B/A + M].$$

Then, for $x \in I$, we have $Q(-B/A) \leq Q(x) \leq Q(-B/A + M)$. Therefore to minimize the absolute value of $Q(x)$ we ask that $Q(-B/A) \approx -Q(-B/A+M)$, which is equivalent to $A^2M^2 \approx 2(B^2 - AC)$ i.e. to

$$A \approx \frac{\sqrt{2(B^2 - AC)}}{M}$$

and we will have

$$\max_{x \in I} |Q(x)| \approx \frac{B^2 - AC}{A} \approx M\sqrt{(B^2 - AC)/2}\,.$$

Since we want this to be as small as possible, but still have $N \mid B^2 - AC$, we will choose A, B and C such that $B^2 - AC = N$ itself, and the maximum of $|Q(x)|$ will then be approximately equal to $M\sqrt{N/2}$.

This is of the same order of magnitude (in fact even slightly smaller) than the size of the values of our initial polynomial $Q(x)$, but now we have the added freedom to change polynomials as soon as the size of the residues become too large for our taste.

To summarize, we first choose an appropriate sieving length M. Then we choose A close to $\sqrt{2N}/M$ such that A is prime and $\left(\frac{N}{A}\right) = 1$. Using Algorithm 1.5.1 we find B such that $B^2 \equiv N \pmod{A}$ and finally we set $C = (B^2 - N)/A$.

Now as in the ordinary quadratic sieve, we must compute for each prime power p^k in our factor base the values $a_{p^k}(Q)$ and $b_{p^k}(Q)$ with which we will initialize our sieve. These are simply the roots mod p^k of $Q(a) = 0$. Hence, since the discriminant of Q has been chosen equal to N, they are equal to $(-B + a_{p^k})/A$ and $(-B + b_{p^k})/A$, where a_{p^k} and b_{p^k} denote the square roots of N modulo p^k which should be computed once and for all. The division by A (which is the only time-consuming part of the operation) is understood modulo p^k.

As for the basic quadratic sieve, heuristically the expected running time of MPQS is $O(L(N)^{1+o(1)})$, as for the class group method and ECM. However, as already mentioned above, the basic operation being so simple, MPQS is much faster than these other methods on numbers which are difficult to factor (numbers equal to a product of two primes having the same order of magnitude).

10.4.3 Improvements to the MPQS Algorithm

The detailed aspects of the implementation of the MPQS algorithm, such as the choice of the sieving intervals, the size of the factor base and criteria to switch from one polynomial to the next, are too technical to be given here. We refer the interested reader to [Sil1] which contains all the necessary information for a well tuned implementation of this algorithm.

A number of improvements can however be mentioned. We have already discussed above the large prime variation. Other improvements are as follows.

(1) One improvement is the double large prime variation. This means that we allow the unfactored part of the residues to be equal not only to a single prime, but also to a product of two primes of reasonable size. This idea is a natural one, but it is then more difficult to keep track of the true relations that are obtained, and A. Lenstra and M. Manasse have found a clever way of doing this. I refer to [LLMP] for details.

(2) A second improvement is the small prime variation which is as follows. During the sieving process, the small primes or prime powers take a very long time to process since about $1/p$ numbers are divisible by p. In addition, their contribution to the logarithms is the smallest. So we do not sieve at all with prime powers less than 100, say. This makes it necessary keep numbers whose residual logarithm is further away from zero than usual, but practice shows that it makes little difference. The main thing is to avoid missing any numbers which factor, at the expense of having a few extra which do not.

(3) A third improvement is the self-initialization procedure. This is as follows. We could try changing polynomials extremely often, since this would be the best chance that the residues stay small, hence factor. Unfortunately, as we have mentioned above, each time the polynomial is changed we must "reinitialize" our sieve, i.e. recompute starting values $a_{p^k}(Q)$ and $b_{p^k}(Q)$ for each p^k in our factor base. Although all the polynomials have the same discriminant N and the square roots have been precomputed (so no additional square root computations are involved), the time-consuming part is to invert the leading coefficient A modulo each element of the factor base. This prevents us from changing polynomial too often since otherwise this would dominate the running time.

The self-initialization procedure deals with this problem by choosing A not to be a prime, but a product of a few (say 10) distinct medium-sized primes p such that $\left(\frac{N}{p}\right) = 1$. The number of possible values for B (hence the number of polynomials with leading term A) is equal to the number of solutions of $B^2 \equiv N \pmod{A}$, and this is equal to 2^{t-1} if t is the number of prime factors of A (see Exercise 30 of Chapter 1). Hence this procedure essentially divides by 2^{t-1} most of the work which must be done in initializing the sieve.

10.5 The Number Field Sieve

10.5.1 Introduction

We now come to the most recent and potentially the most powerful known factoring method, the number field sieve (NFS). For complete details I refer to [Len-Len2]. The basic idea is the same as in the quadratic sieve: by a sieving process we look for congruences modulo N by working over a factor base, and then we do a Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$ to obtain a congruence of squares, hence hopefully a factorization of N.

Before describing in detail the method, we will comment on its performance. Prior to the advent of the NFS, all modern factoring methods had an expected running time of at best $O(e^{\sqrt{\ln N \ln\ln N}(1+o(1))})$. Because of the theorem of Canfield, Erdős and Pomerance, some people believed that this could not be improved, except maybe for the $(1+o(1))$. The invention by Pollard of the NFS has now changed this belief, since under reasonable heuristic assumptions, one can show that the expected running time of the NFS is

$$O\left(e^{(\ln N)^{1/3}(\ln\ln N)^{2/3}(C+o(1))}\right)$$

for a small constant C (an admissible value is $C = (64/9)^{1/3}$ and this has been slightly lowered by Coppersmith). This is asymptotically considerably better than what existed before. Unfortunately, the practical situation is less simple. First, for a number N having no special form, it seems that the practical cutoff point with, say, the MPQS method, is for quite large numbers, maybe around 130 digits, and these numbers are in any case much too large to be factored by present methods. On the other hand, for numbers having a special form, for example Mersenne numbers $2^p - 1$ or Fermat numbers $2^{2^k} + 1$, NFS can be considerably simplified (one can in fact decrease the constant C to $C = (32/9)^{1/3}$), and stays practical for values of N up to 120 digits. In fact, using a system of distributed e-mail computing, and the equivalent of years of CPU time on small workstations, A. K. Lenstra and Manasse succeeded in 1990 in factoring the ninth Fermat number $F_9 = 2^{512} + 1$, which is a number of 155 decimal digits. The factors have respectively 7, 49 and 99 digits and the 7-digit factor was of course already known. Note that the knowledge of this 7-digit factor does not help NFS at all in this case.

The idea of the number field sieve is as follows. We choose a number field $K = \mathbb{Q}(\theta)$ for some algebraic integer θ, let $T(X) \in \mathbb{Z}[X]$ be the minimal monic polynomial of θ, and let d be the degree of K. Assume that we know an integer m such that $T(m) = kN$ for a small integer k. Then we can define a ring homomorphism ϕ from $\mathbb{Z}[\theta]$ to $\mathbb{Z}/N\mathbb{Z}$, by setting

$$\phi(\theta) = m \bmod N.$$

This homomorphism can be extended to $\mathbb{Z}_K$ in the following way. Let $f = [\mathbb{Z}_K : \mathbb{Z}[\theta]]$ be the index of $\mathbb{Z}[\theta]$ in $\mathbb{Z}_K$. We may assume that $(f, N) = 1$

otherwise we have found a non-trivial factor of N. Hence f is invertible modulo N, and if $u \in \mathbb{Z}$ is an inverse of f modulo N, for all $\alpha \in \mathbb{Z}_K$ we can set $\phi(\alpha) = u\phi(f\alpha)$ since $f\alpha \in \mathbb{Z}[\theta]$.

We can use ϕ as follows. To take the simplest example, if we can find integers a and b such that $a + bm$ is a square (in $\mathbb{Z}$), and also such that $a + b\theta$ is a square (in $\mathbb{Z}_K$), then we may have factored N: write $a + bm = x^2$, and $a + b\theta = \beta^2$. Since ϕ is a ring homomorphism, $\phi(a + b\theta) = a + bm \equiv y^2 \pmod{N}$ where we have set $y \pmod{N} = \phi(\beta)$, hence $x^2 \equiv y^2 \pmod{N}$, so $(x - y, N)$ may be a non-trivial divisor of N. Of course, in practice it will be impossible to obtain such integers a and b directly, but we can use techniques similar to those which we used in the continued fraction or in the quadratic sieve method, i.e. factor bases. Here however the situation is more complicated. We can take a factor base consisting of primes less than a given bound for the $a + bm$ numbers. But for the $a + b\theta$, we must take prime *ideals* of $\mathbb{Z}_K$. In general, if K is a number field with large discriminant, this will be quite painful. This is the basic distinction between the general number field sieve and the special one: if we can take for K a simple number field (i.e. one for which we know everything: units, class number, generators of small prime ideals, etc ...) then we are in the special case.

We will start by describing the simplest case of NFS, which can be applied only to quite special numbers, and in the following section we will explain what must be done to treat numbers of a general form.

10.5.2 Description of the Special NFS when $h(K) = 1$

In this section we not only assume that K is a simple number field in the sense explained above, but in addition that $\mathbb{Z}_K$ has class number equal to 1 (we will see in the next section what must be done if this condition is not satisfied).

Let $\alpha \in \mathbb{Z}_K$ and write

$$\alpha\mathbb{Z}_K = \prod_i \mathfrak{p}_i^{v_i},$$

where we assume that for all i, $v_i > 0$. We will say that α is B-smooth if $\mathcal{N}_{K/\mathbb{Q}}(\alpha)$ is B-smooth, or in other words if all the primes below $\mathfrak{p}_i$ are less than or equal to B. Since $\mathbb{Z}_K$ has class number equal to 1, we can write

$$\alpha = \prod_{u \in U} u^{\lambda_u} \prod_{g \in G} g^{\mu_g},$$

where U is a generating set of the group of units of K (i.e. a system of fundamental units plus a generator of the subgroup of roots of unity in K), and G is a set of $\mathbb{Z}_K$-generators for the prime ideals $\mathfrak{p}$ above a prime $p \leq B$ (since the ideals $\mathfrak{p}$ are principal).

If a lift of $\phi(\alpha)$ to $\mathbb{Z}$ is also B-smooth (in practice we always take the lift in $[-N/2, N/2]$) then we have

$$\phi(\alpha) \equiv \prod_{p \le B} p^{v_p}$$

hence the congruence

$$\prod_{u \in U} \phi(u)^{\lambda_u} \prod_{g \in G} \phi(g)^{\mu_g} \equiv \prod_{p \le B} p^{v_p} \pmod{N}.$$

If $\mathcal{P}$ is the set of primes less than or equal to B, then as in the quadratic sieve and similar algorithms, if we succeed in finding more than $|U|+|G|+|\mathcal{P}|$ such congruences, we can factor N by doing Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$.

By definition an HNF basis of $\mathbb{Z}_K$ is of the form $(1, (u\theta + v)/w, \dots)$. Replacing, if necessary, θ by $(u\theta + v)/w$, without loss of generality we may assume that there exists an HNF basis of $\mathbb{Z}_K$ of the form $(\omega_1, \omega_2, \omega_3, \dots, \omega_d)$ where $\omega_1 = 1$, $\omega_2 = \theta$ and ω_i is of degree exactly equal to $i-1$ in θ. We will say in this case that θ is *primitive*.

This being done, we will in practice choose α to be of the form $a + b\theta$ with a and b in $\mathbb{Z}$ and coprime. We have the following lemma.

Lemma 10.5.1. *If a and b are coprime integers, then any prime ideal $\mathfrak{p}$ which divides $a + b\theta$, either divides the index $f = [\mathbb{Z}_K : \mathbb{Z}[\theta]]$ or is of degree 1.*

Proof. Let p be the prime number below $\mathfrak{p}$. Then $p \nmid b$ otherwise $a \in \mathfrak{p} \cap \mathbb{Z}$ hence $p \mid a$, contradicting a and b being coprime. Now assume that $p \nmid f$, and let b^{-1} be an inverse of b modulo p and u be an inverse of f modulo p. We have $\theta \equiv -ab^{-1} \pmod{\mathfrak{p}}$. Hence, if $x \in \mathbb{Z}_K$, $fx \in \mathbb{Z}[\theta]$ so there exists a polynomial $P \in \mathbb{Z}[X]$ such that $x \equiv uP(-ab^{-1}) \pmod{\mathfrak{p}}$ so any element of $\mathbb{Z}_K$ is congruent to a rational integer modulo $\mathfrak{p}$, hence to an element of the set $\{0, 1, \dots, p-1\}$, thus proving the lemma. □

Let $d = \deg(T)$ be the degree of the number field K. By Theorem 4.8.13, prime ideals of degree 1 dividing a prime number p not dividing the index correspond to linear factors of $T(X)$ modulo p, i.e. to roots of $T(X)$ in $\mathbb{F}_p$. These can be found very simply by using Algorithm 1.6.1.

For any root $c_p \in \{0, 1, \dots, p-1\}$ of $T(X)$ modulo p, we thus have the corresponding prime ideal of degree 1 above p generated over $\mathbb{Z}_K$ by $(p, \theta - c_p)$. Now when we factor numbers α of the form $a + b\theta$ with $(a, b) = 1$, we will need to know the $\mathfrak{p}$-adic valuation of α for all prime ideals $\mathfrak{p}$ such that $\alpha \in \mathfrak{p}$. But clearly, if p does not divide f, then $\alpha \in \mathfrak{p}$ if and only if $p \mid a + bc_p$, and if this is the case then α does not belong to any other prime above p since the c_p are distinct. Hence, if $p \mid a + bc_p$, the $\mathfrak{p}$-adic valuation of α (with $\mathfrak{p} = (p, \theta - c_p)$) is equal to the p-adic valuation of $\mathcal{N}(\alpha)$ which is simple to compute.

For $p \mid f$, we can use an HNF basis of $\mathfrak{p}$ with respect to θ, where we may assume that θ is primitive. This basis will then be of the form $(p, -c_p + y\theta, \gamma_2, \dots, \gamma_{d-1})$ where c_p and y are integers with $y \mid p$ and the

γ_i are polynomials of degree exactly i in θ (not necessarily with integral coefficients). It is clear that $a+b\theta \in \mathfrak{p}$ if and only if $y \mid b$ and $a \equiv -bc_p/y \pmod{p}$. But $p \mid b$ is impossible since as before it would imply $p \mid a$ hence a and b would not be coprime. It follows that we must have $y = 1$. Hence, $\alpha \in \mathfrak{p}$ if and only if $p \mid a+bc_p$. Furthermore, $\theta - c_p \in \mathfrak{p}$ implies clearly that $T(c_p) \equiv 0 \pmod{p}$, i.e. that c_p is a root of T modulo p. The condition is therefore exactly the same as in the case $p \nmid f$. Note however that now there may be several prime ideals $\mathfrak{p}$ with the same value of c_p, so in that case the $\mathfrak{p}$-adic valuation of α should be computed using for example Algorithm 4.8.17. (Since this will be done only when we know that α and $\phi(\alpha)$ are B-smooth, it does not matter in practice that Algorithm 4.8.17 takes longer than the computation of $v_p(\mathcal{N}(\alpha))$.)

Thus, we will compute once and for all the roots c_p of the polynomial $T(X)$ modulo each prime $p \leq B$, and the constants $\beta_{\mathfrak{p}}$ (β in the notation of Algorithm 4.8.17) necessary to apply directly step 3 of Algorithm 4.8.17 for each prime ideal $\mathfrak{p}$ dividing the index. It is then easy to factor $\alpha = a + b\theta$ into prime *ideals* as explained above. Note that in the present situation, it is not necessary to split completely the polynomial $T(X)$ modulo p using one of the methods explained in Chapter 3, but only to find its roots modulo p, and in that case Algorithm 1.6.1 is much faster.

We must however do more, that is we need to factor α into prime *elements* and units. This is more delicate.

First, we will need to find explicit generators of the prime ideals in our factor base (recall that we have assumed that $\mathbb{Z}_K = \mathbb{Z}[\theta]$ is a PID). This can be done by computing norms of a large number of elements of $\mathbb{Z}_K$ which can be expressed as polynomials in θ with small coefficients, and combining the norms to get the desired prime numbers. This operation is quite time consuming, and can be transformed into a probabilistic algorithm, for which we refer to [LLMP]. This part is the essential difference with the general NFS since in the general case it will be impossible in practice to find generators of principal ideals. (The fact that $\mathbb{Z}_K$ is not a PID in general also introduces difficulties, but which are less important.)

Second, we also need generators for the group of units. This can be done during the search for generators of prime ideals. We find in this way a generating system for the units, and the use of the complex logarithmic embedding allows us to extract a multiplicative basis for the units as in Algorithm 6.5.9.

Choosing a factor base limit B, we will take as factor base for the numbers $a+bm$ the primes p such that $p \leq B$, and as factor base for the numbers $a+b\theta$ we will take a system G of non-associate prime *elements* of $\mathbb{Z}_K$ whose norm is either equal to $\pm p$, where p is a prime such that $p \leq B$ and $p \nmid f$, or equal to $\pm p^k$ for some k if $p \leq B$ and $p \mid f$, plus a generating system of the group of units of $\mathbb{Z}_K$.

We have seen that $\alpha \in \mathfrak{p}$ if and only if $p \mid a + bc_p$ which is a linear congruence for a and b. Hence, we can sieve using essentially the same sieving procedure as the one that we have described for the quadratic sieve.

1) By sieving on small primes, eliminate pairs (a, b) divisible by a small prime. (We will therefore keep a few pairs with $(a, b) > 1$, but this will not slow down the procedure in any significant way.)

2) Initialize the entries in the sieving interval to a crude approximation to $\ln(a + mb)$.

3) First sieve: for every $p^k \leq B$, subtract $\ln p$ from the entries where $p^k \mid a + mb$ by sieving modulo p, p^2, ...

4) Set a flag on all the entries which are still large (i.e. which are not B-smooth), and initialize the other entries with $\ln(\mathcal{N}(a + b\theta))$.

5) Second sieve: for every pair (p, c_p), subtract $\ln p$ from the unflagged entries for which $p \mid a + bc_p$. Note that we cannot sieve modulo p^2, ...

6) For each entry which is smaller than $2 \ln B$ (say), check whether the corresponding $\mathcal{N}(a+b\theta)$ is indeed smooth and in that case compute the complete factorization of $a+b\theta$ on $G \cup U$. Note that since we have not sieved with powers of prime ideals, we must check some entries which are larger than $\ln B$.

In practice, the factorization of $a+b\theta$ is obtained as follows. Since $\mathcal{N}(a+b\theta)$ is smooth we know that $\mathcal{N}(a + b\theta) = \prod_{p \leq B} p^{v_p}$. We can obtain the element relations as follows. If only one prime ideal $\mathfrak{p}$ above p corresponds to a given c_p (this is always true if $p \nmid f$), then if we let d be the degree of $\mathfrak{p}$ (1 if $p \nmid f$), the $\mathfrak{p}$-adic valuation of $a + b\theta$ is v_p/d, and the $\mathfrak{p}'$-adic valuation is zero for every other prime ideal above p. If several prime ideals correspond to the same c_p (this is possible only in the case $p \mid f$), then we use Algorithm 4.8.17 to compute the $\mathfrak{p}$-adic valuations. As already mentioned, this will be done quite rarely and does not really increase the running time which is mainly spent in the sieving process. Using the set G of explicit generators of our prime ideals, we thus obtain a decomposition

$$a + b\theta = u \prod_{g \in G} g^{\mu_g}$$

where u is a unit. If $(u_1, \ldots, u_r)$ is a system of fundamental units of K and ζ is a generator of the group of roots of unity in K, we now want to write

$$u = \zeta^{n_0} \prod_{i=1}^{r} u_i^{n_i}.$$

To achieve this, we can use the logarithmic embedding L (see Definition 4.9.6) and compute $L(a+b\theta) - \sum_{g \in G} \mu_g L(g)$. This will lie in the hyperplane $\sum x_i = 0$ of $\mathbb{R}^{r_1+r_2}$, and by Dirichlet's theorem, the $L(u_i)$ form a basis of this hyperplane, hence we can find the n_i for $i \geq 1$ by solving a linear system (over $\mathbb{R}$, but we know that the solution is integral). Finally, n_0 can be obtained by comparing arguments of complex numbers (or even more simply by comparing signs if everything is real, which can be assumed if d is odd).

10.5.3 Description of the Special NFS when $h(K)>1$

In this section, we briefly explain what modifications should be made to the above method in the case $h(K) > 1$, hence when $\mathbb{Z}_K$ is not a PID.

In this case we do not try to find generators of the prime ideals, but we look as before for algebraic integers (not necessarily of the form $a + b\theta$) with small coordinates in an integral basis, having a very smooth norm. More precisely, let $\mathfrak{p}_1$, $\mathfrak{p}_2$, ... be the prime ideals of norm less than or equal to B ordered by increasing norm. We first look for an algebraic integer a_1 whose decomposition gives $a_1\mathbb{Z}_K = \mathfrak{p}_1^{k_{1,1}}$ where $k_{1,1}$ is minimal and hence is equal to the order of $\mathfrak{p}_1$ in $Cl(K)$. Then we look for another algebraic integer a_2 such that $a_2\mathbb{Z}_K = \mathfrak{p}_1^{k_{1,2}}\mathfrak{p}_2^{k_{2,2}}$ where $k_{2,2}$ is minimal and hence is equal to the order of $\mathfrak{p}_2$ in $Cl(K)/ < \mathfrak{p}_1 >$. We may also assume that $k_{1,2} < k_{1,1}$. We proceed in this way for each $\mathfrak{p}_i$ of norm less than or equal to B, and thus we have constructed an upper triangular matrix M whose rows correspond to the prime ideals and whose columns correspond to the numbers a_i. With high probability we have $h(K) = \prod_i k_{i,i}$, but it does not matter if this is not the case.

We can now replace the set G of generators of the $\mathfrak{p}_i$ which was used in the case $h(K) = 1$ by the set of numbers a_i in the following way.

Assume that α is B-smooth and that $\alpha\mathbb{Z}_K = \prod_i \mathfrak{p}_i^{v_i}$. Let V be the column vector whose components are the v_i. It is clear that $\alpha\mathbb{Z}_K = \prod_j a_j^{\mu_j}\mathbb{Z}_K$ where the μ_j are the components of the vector $M^{-1}V$ which are integers by construction of the matrix M. Hence $\alpha = u\prod_j a_j^{\mu_j}$ where u is a unit, and we can proceed as before. Note that since M is an upper triangular matrix it is easy to compute $M^{-1}V$ by induction.

An Example of the Special NFS. Assume that N is of the form $r^e - s$, where r and s are small. Choose a suitable degree d ($d = 5$ is optimal for numbers having 70 digits or more), and set $k = \left\lceil \frac{e}{d} \right\rceil$. Consider the polynomial

$$T(X) = X^d - sr^{kd-e}.$$

Since $0 \leq kd - e < d$ and s and r are small, so is sr^{kd-e}. If we choose $m = r^k$, it is clear that $T(m) = r^{kd-e}N$ is a small multiple of N. If T is an irreducible polynomial, we will work in the number field K of degree d defined by T. (If T is reducible, which almost never happens, we usually obtain a non-trivial factorization of N from a non-trivial factorization of T.) Since typically $d = 5$, and sr^{kd-e} is small, K is a simple field, i.e. it will not be difficult to find generators for ideals of small norm, the class number and a generating system for the group of units.

As mentioned above, the first success of the special NFS was obtained by [LLMP] with the ninth Fermat number $N = 2^{512} + 1$ which is of the above form. They chose $d = 5$, hence $k = 103$ and $T(X) = X^5 + 8$, thus $K = \mathbb{Q}(2^{1/5})$ which happens to be a field with class number equal to 1.

10.5.4 Description of the General NFS

The initial ideas of the general NFS are due to Buhler and Pomerance (see [BLP]). We do not assume anymore that K is a simple field. Hence it is out of the question to compute explicit generators for prime ideals of small norm, a system of fundamental units, etc ... Hence, we must work with ideals (and not with algebraic numbers) as long as possible.

So we proceed as before, but instead of keeping relations between elements (which is not possible anymore), we keep relations between the prime ideals themselves. As usual in our factor base we take the prime ideals of degree 1 whose norm is less than or equal to B and the prime ideals of norm less than or equal to B which divide the index f; since the index may not be easy to compute, we can use instead the prime ideals above primes $p \leq B$ such that p^2 divides the discriminant of the polynomial T).

After the usual Gaussian elimination step over $\mathbb{Z}/2\mathbb{Z}$, we will obtain algebraic numbers of the form

$$y = \prod (a + b\theta)^{\varepsilon_{a,b}}$$

where without loss of generality we may assume that $\varepsilon_{a,b} = 0$ or 1, such that

$$\phi(y) = \prod_{p \leq B} p^{v_p} \quad \text{(i.e. } \phi(y) \text{ is } B\text{-smooth), and}$$

$$y\mathbb{Z}_K = \prod_{\mathfrak{p}} \mathfrak{p}^{2v_{\mathfrak{p}}},$$

this last product being over the prime ideals of our factor base. Although the principal ideal $y\mathbb{Z}_K$ is equal to the square of an ideal, this does not imply that it is equal to the square of a *principal* ideal. Fortunately, this difficulty can easily be overcome by using a trick due to L. Adleman (see [Adl]).

Let us say that a non-zero algebraic number $y \in K$ is *singular* if $y\mathbb{Z}_K$ is the square of a fractional ideal. Let S be the multiplicative group of singular numbers. If $U(K)$ is the group of units of K, it is easy to check that we have an exact sequence

$$1 \longrightarrow U(K)/U(K)^2 \longrightarrow S/K^{*2} \longrightarrow Cl(K)[2] \longrightarrow 1,$$

where for any Abelian group G, $G[2]$ is the subgroup of elements of G whose square is equal to the identity (see Exercise 9). This exact sequence can be considered as an exact sequence of vector spaces over $\mathbb{F}_2 = \mathbb{Z}/2\mathbb{Z}$. Furthermore, using Dirichlet's Theorem 4.9.5 and the parity of the number $w(K)$ of roots of unity in K, it is clear that

$$\dim_{\mathbb{F}_2} U(K)/U(K)^2 = r_1 + r_2.$$

For any finite Abelian group G, the exact sequence

$$1 \longrightarrow G[2] \longrightarrow G \longrightarrow G \longrightarrow G/G^2 \longrightarrow 1,$$

where the map from G to G is squaring, shows that $|G[2]| = |G/G^2|$ hence

$$\dim_{\mathbb{F}_2} G[2] = rk_2(G),$$

where the 2-rank $rk_2(G)$ of G is by definition equal to $\dim_{\mathbb{F}_2} G/G^2$ (and also to the number of even factors in the decomposition of G into a direct product of cyclic factors). Putting all this together, we obtain

$$\dim_{\mathbb{F}_2}(S/K^{*2}) = r_1 + r_2 + rk_2(Cl(K)).$$

Hence, if we obtain more than $e = r_1+r_2+rk_2(Cl(K))$ singular numbers which are algebraic integers, a suitable multiplicative combination with coefficients 0 or 1 will give an element of $\mathbb{Z}_K \cap K^{*2}$, i.e. a square of $\mathbb{Z}_K$, as in the special NFS, hence a true relation of the form we are looking for. Since e is very small, this simply means that instead of stopping at the first singular integer that we find, we wait till we have at least $e+1$ more relations than the cardinality of our factor base. Note that it is *not* necessary (and in practice not possible) to compute $rk_2(Cl(K))$. Any guess is sufficient, since afterwards we will have to check that we indeed obtain a square with a suitable combination, and if we do not obtain a square, this simply means that our guess is not large enough.

To find a suitable combination, following Adleman we proceed as follows. Choose a number r of prime ideals $\mathfrak{p}$ which do not belong to our factor base. A reasonable choice is $r = 3e$, where e can (and must) be replaced by a suitable upper bound. For example, we can choose for $\mathfrak{p}$ ideals of degree 1 above primes which are larger than B. Then $\mathfrak{p} = (p, \theta - c_p)$. We could also choose prime ideals of degree larger than 1 above primes (not dividing the index) less than B.

Whatever choice is made, the idea is then to compute a generalized Legendre symbol $\left(\frac{a+b\theta}{\mathfrak{p}}\right)$ (see Exercise 19 of Chapter 4) for every $a+b\theta$ which is kept after the sieving process. Hence each relation will be stored as a vector over $\mathbb{Z}/2\mathbb{Z}$ with $|E| + |\mathcal{P}| + r$ components, where E is the set of prime ideals in our factor base. As soon as we have more relations than components, by Gaussian elimination over $\mathbb{Z}/2\mathbb{Z}$ we can find an algebraic number x which is a singular integer and which is a quadratic residue modulo our r extra primes $\mathfrak{p}$. It follows that x is quite likely a square.

Assuming this to be the case, one of the most difficult problems of the general number field sieve, which is not yet satisfactorily solved at the time of this writing, is the problem of finding an algorithm to compute a square root y of x. Note that in practice x will be a product of thousands of $a+b\theta$, hence will be an algebraic number with coefficients (as polynomials in θ, say) having several hundred thousand decimal digits. Although feasible in principle, it does not seem that the explicit computation of x as a polynomial in θ will be of much help because of the size of the coefficients involved. Similarly for any other practical representation of x, for example by its minimal polynomial.

Let us forget this practical difficulty for the moment. We would like an algorithm which, given an algebraic integer x of degree d, either finds $y \in \mathbb{Q}[x]$ such that $y^2 = x$, or says that such a y does not exist. A simple-minded algorithm to achieve this is as follows.

Algorithm 10.5.2 (Square Root in a Number Field). Given an algebraic integer x by its minimal monic polynomial $A(X) \in \mathbb{Z}[X]$ of degree d, this algorithm finds a y such that $y^2 = x$ and $y \in \mathbb{Q}[x]$, or says that such a y does not exist. (If x is given in some other way than by its minimal polynomial, compute the minimal polynomial first.) We let $K = \mathbb{Q}[x]$.

1. [Factor $A(X^2)$] Factor the polynomial $A(X^2)$ in $\mathbb{Z}[X]$. If $A(X^2)$ is irreducible, then y does not exist and terminate the algorithm. Otherwise, let $A(X^2) = \pm S(X)S(-X)$ for some monic polynomial $S \in \mathbb{Z}[X]$ of degree d be the factorization of $A(X^2)$ (it is necessarily of this form with S irreducible, see Exercise 10).
2. [Reduce to degree 1] Let $S(X) = (X^2 - x)Q(X) + R(X)$ be the Euclidean division of $S(X)$ by $X^2 - x$ in $K[X]$.
3. [Output result] Write $R(X) = aX + b$ with a and b in K and $a \neq 0$. Output $y \leftarrow -b/a$ and terminate the algorithm.

The proof of the validity of this algorithm is easy and left to the reader (Exercise 10).

Unfortunately, in our case, simply computing the polynomial $A(X)$ is already not easy, and factoring $A(X^2)$ will be even more difficult (although it will be a polynomial of degree 10 for example, but with coefficients having several hundred thousand digits). So a new idea is needed at this point. For example, H. W. Lenstra has suggested looking for y of the form $y = \prod(a+b\theta)$, the product being over coprime pairs (a, b) such that $a + b\theta$ is smooth, but not necessarily $a + bm$. This has the advantage that many more pairs (a, b) are available, and also leads to a linear system over $\mathbb{Z}/2\mathbb{Z}$. Future work will tell whether this method or similar ones are sufficiently practical.

10.5.5 Miscellaneous Improvements to the Number Field Sieve

Several improvements have been suggested to improve the (theoretical as well as practical) performance of NFS. Most of the work has been done on the general NFS, since the special NFS seems to be in a satisfactory state. We mention only two, since lots of work is being done on this subject.

The most important choice in the general NFS is the choice of the number field K, i.e. of the polynomial $T \in \mathbb{Z}[X]$ such that $T(m) = kN$ for some small integer k. Choosing a fixed degree d (as already mentioned, $d = 5$ is optimal for numbers having more than 60 or 70 digits), we choose $m = \lfloor N^{1/d} \rfloor$. If $N = m^d + a_{d-1}m^{d-1} + \cdots + a_0$ is the base m expansion of N (with $0 \leq a_i < m$), we can choose

$$T(X) = X^d + a_{d-1}X^{d-1} + \cdots + a_0.$$

It is however not necessary to take the base m expansion of N in the strictest sense, since any base m expansion of N whose coefficients are at most of the order of m is suitable. In addition, we can choose to expand some small multiple kN of N instead of N itself. This gives us additional freedom.

Another idea is to use $m = \lceil N^{1/(d+1)} \rceil$ instead of $\lfloor N^{1/d} \rfloor$. The base m expansion of N is then of the form $N = a_d m^d + a_{d-1}m^{d-1} + \cdots + a_0$ with a_d not necessarily equal to 1, but still less than m. We take as before

$$T(X) = a_d X^d + a_{d-1}X^{d-1} + \cdots + a_0,$$

and if θ is a root of T, then θ is not an algebraic integer if $a_d > 1$. We can now use Exercise 15 of Chapter 4 which tells us that $a_d\theta$, $a_d\theta^2 + a_{d-1}\theta$, ... are algebraic integers. The map ϕ is defined as usual by $\phi(\theta) = m$ and extended to polynomials in θ with integer coefficients. In particular, if a and b are integers, $a_d(a + b\theta)$ is an algebraic integer and

$$\phi(a_d(a + b\theta)) = a_d(a + mb)$$

is always divisible by a_d. Also,

$$\mathcal{N}(a_d(a + b\theta)) = (-1)^d a_d^{d-1} b^d T(-a/b)$$

with $b^d T(-a/b) \in \mathbb{Z}$. We then proceed as before, but using numbers of the form $a_d(a + b\theta)$ with a and b coprime, instead of simply $a + b\theta$.

To get rid of a_d in the final relations, it is not necessary to include the prime factors of a_d in the factor base, but simply to take an even number of factors in each relation.

A second type of improvement, studied by D. Coppersmith, is to use *several* number fields K. This leads to an improvement of the constant in the exponent of the running time of NFS, but its practicality has not yet been tested. The idea is a little similar to the use of several polynomials in MPQS.

10.6 Exercises for Chapter 10

1. Show that the problem of computing a square root modulo an arbitrary integer N is probabilistically polynomial time equivalent to the problem of factoring N in the following sense. If we have an algorithm for one of the problems, then we can solve the other in probabilistic polynomial time.
2. Generalize Algorithm 10.2.2 by incorporating a second stage in the manner of Algorithm 8.8.3.
3. Show how to write the addition law on an elliptic curve modulo N given by a Weierstraß equation using projective coordinates, using 12 multiplications modulo N, or 13 for the double of a point.

4. By using a Fermat parametrization of an elliptic curve, i.e. a projective equation of the form $x^3 + ay^3 = bt^3$, show how to compute the addition law using only 9 multiplications modulo N, or 10 for the double of a point.

5. Let B and k be large integers, and let $a_1, \ldots a_k$ be a randomly chosen sequence of integers less than B. Give an estimate of the average number of pairs (i, j) such that $a_i = a_j$. You may assume that $k > B^{1/2}$.

6. Let n be fixed, and set $f(N) = |\mathbb{P}_n(\mathbb{Z}/N\mathbb{Z})|$.
   a) Show that $f(N) = g(N)/\phi(N)$ where $\phi(N)$ is the Euler ϕ function, and $g(N)$ is the number of $n+1$-uples $(x_0, \ldots, x_n) \in (\mathbb{Z}/N\mathbb{Z})^{n+1}$ such that $\gcd(x_0, \ldots, x_n, N) = 1$.
   b) Show that $\sum_{d|N} g(d) = N^{n+1}$.
   c) Using the Möbius inversion formula (see [H-W] Section 16.4), prove the formula for $f(N)$ given in the text.

7. In the multiple polynomial version of the quadratic sieve factoring algorithm, we have $aQ(x) \equiv y^2 \pmod{N}$ for some N, and not $Q(x)$ itself. Then why do we take into account in the explanation the maximum of $|Q(x)|$ and not of $|aQ(x)|$?

8. Let $\mathfrak{p} = (p, \theta - c_p)$ be a prime ideal of degree 1 in $\mathbb{Z}_K$, where $K = \mathbb{Q}(\theta)$. If $x = a + b\theta \in \mathbb{Z}_K$, show that $\left(\frac{x}{\mathfrak{p}}\right) = \left(\frac{a+bc_p}{p}\right)$, where $\left(\frac{x}{\mathfrak{p}}\right)$ is defined in Exercise 19 of Chapter 4.

9. Prove that, as claimed in the text, if S is the group of singular numbers, the following sequence is exact:

$$1 \longrightarrow U(K)/U(K)^2 \longrightarrow S/K^{*2} \longrightarrow Cl(K)[2] \longrightarrow 1,$$

   where $Cl(K)[2]$ is the subgroup of elements of $Cl(K)$ whose square is equal to the identity.

10. Let $A(X)$ be an irreducible monic polynomial in $\mathbb{Z}[X]$.
    a) Show that either $A(X^2)$ is irreducible in $\mathbb{Z}[X]$, or there exists an irreducible monic polynomial $S \in \mathbb{Z}[X]$ such that $A(X^2) = \pm S(X)S(-X)$.
    b) Prove the validity of Algorithm 10.5.2.

11. For any finite Abelian group G and $n \geq 1$ show that

$$G[n] \simeq G/G^n$$

    (although this isomorphism is not canonical in general).

Appendix A

Packages for Number Theory

There exist several computer packages which can profitably be used for number-theoretic computations. In this appendix, I will briefly describe the advantages and disadvantages of some of these systems.

Most general-purpose symbolic algebra packages have been written primarily for applied mathematicians, engineers and physicists, and are not always well suited for number theory. These packages roughly fall into two categories. In the first category one finds computer algebra systems developed in the 1970's, of which the main representatives are Macsyma and Reduce. Because of their maturity, these systems have been extensively tested and have probably less bugs than more recent systems. In addition they are very often mathematically more robust. In the second category, I include more recent packages developed in the 1980's of which the most common are Mathematica, by Wolfram Research, Inc., Maple, by the University of Waterloo, Canada, and more recently Axiom, developed by IBM and commercialized by NAG. These second-generation systems being more recent have more bugs and have been less tested. They are also often more prone to mathematical errors. On the other hand they have been aggressively commercialized and as a consequence have become more popular. However, the older systems have also been improved, and in particular recently Macsyma was greatly improved in terms of speed, user friendliness and efficiency and now compares very favorably to more recent packages. Mathematica has a very nice user interface, and its plotting capabilities, for example on the Macintosh, are superb. Maple is faster and often simpler to use, and has my preference. Axiom is a monster (in the same sense that ADA is a monster as a programming language). It certainly has a large potential for developing powerful applications, but I do not believe that there is the need for such power (which is usually obtained at the expense of speed) for everyday (number-theoretic) problems.

Some other packages were specially designed for small machines like Personal Computers (PC's). One of these is Derive, which is issued from μ-Math, and requires only half a megabyte of main memory. Derive even runs on some pocket computers! Another system, the Calculus Calculator (CC), is a symbolic manipulator with three-dimensional graphics and matrix operations which also runs on PC's. A third system, Numbers, is a shareware calculator for number theory that runs on PC's. It is designed to compute number theoretic functions for positive integers up to 150 decimal digits (modular

arithmetic, primality testing, continued and Farey fractions, Fibonacci and Lucas numbers, encryption and decryption).

In addition to commercial packages, free software systems (which are not complete symbolic packages) also exist. One is Ubasic, written by Y. Kida, which is a math-oriented high-precision Basic for PC's (see the review in the Notices of the AMS of March 1991). Its extensions to Basic allow it to handle integers and reals of several thousand digits, as well as fractions, complex numbers and polynomials in one variable. Many number-theoretic functions are included in Ubasic, including the factoring algorithm MPQS. Since the package is written in assembly language, Ubasic is very fast.

Another package, closer to a symbolic package, is Pari, written by the author and collaborators (see the review in the Notices of the AMS of October 1991). This package can be used on Unix workstations, Macintosh, Amiga, PC's, etc. Its kernel is also written in assembler, so it is also very fast. Furthermore, it has been specially tailored for number-theoretic computations. In addition, it provides tools which are rarely or never found in other symbolic packages such as the direct handling of concrete mathematical objects, for example p-adic numbers, algebraic numbers and finite fields, etc ... It also gives mathematically more correct results than many packages on fundamental operations (e.g. subtraction of two real numbers which are approximately equal).

Source is included in the package so it is easy to correct, improve and expand. Essentially all of the algorithms described in the present book have been implemented in Pari, so I advise the reader to obtain a copy of it.

Apart from those general computer algebra systems, some special-purpose systems exist: GAP, Kant, Magma, Simath. The Magma system is designed to support fast computations in algebra (groups, modules, rings, polynomial rings over various kinds of coefficient domains), number theory and finite geometry. It includes general machinery for classical number theory (for example the ECM program of A.K. Lenstra), finite fields and cyclotomic fields and facilities for computing in a general algebraic number field. It will eventually include a MPQS factoring algorithm, a Jacobi sum-type primality test and a general purpose elliptic curve calculator. According to the developers, it should eventually include "just about all of the algorithms of this book". GAP (Groups, Algorithms and Programming) is specially designed for computations in group theory. It includes some facilities for doing elementary number theory, in particular to calculate with arbitrary length integers and rational numbers, cyclotomic fields and their subfields, and finite fields. It has functions for integer factorization (based on elliptic curves), for primality testing, and for some elementary functions from number theory and combinatorics. Its programming language is Maple-like. Kant (Komputational Algebraic Number Theory) is a subroutine package for algorithms from the geometry of numbers and algebraic number theory, which will be included in Magma. Simath, developed at the university of Saarbrucken, is another system for number-theoretic computations which is quite fast and has a nice user interface called simcalc.

In addition to specific packages, handling of multi-precision numbers or more general types can be easily achieved with several languages, Lisp, C and C++. For Lisp, the INRIA implementation LeLisp (which is not public domain) contains a package written in assembler to handle large numbers, and hence is very fast. The GNU Calc system is an advanced desk calculator for GNU Emacs, written in Emacs Lisp. An excellent public domain C++ compiler can be obtained from the Free Software Foundation, and its library allows to use multi-precision numbers or other types. The library is however written in C++ hence is *slow*, so it is strongly advised to write a library in assembler for number-theoretic uses. Another multi-precision system written in C is the desk calculator (Calc) of Hans-J. Boehm for Unix workstations. Its particularity is to handle "constructive" real numbers, that is to remember the best known approximation to a number already computed. For PC's, Timothy C. Frenz has developed an "infinite" precision calculator, also named Calc.

Finally, a few free packages exist which have been specifically written for handling multi-precision integers as part of a C library in an efficient way. In addition to Pari mentioned above, there is the Bignum package of DEC PRL (which is essentially the package used in LeLisp as mentioned above) which can be obtained by sending an e-mail message to `librarian@decprl.dec.com`, and the GNU multi-precision package Gmp which can be obtained by anonymous ftp from `prep.ai.mit.edu`, the standard place where one can ftp all the GNU software.

Conclusions.

My personal advice (which is certainly not objective) is the following. If you are on an IBM-PC 286, you do not have much choice. Obtain Ubasic, Derive or the Calculus Calculator. On an IBM-PC 386 or more, Maple, Macsyma, Mathcad (see Maple below) and Pari are also available. If you are on a MacII or on a Unix workstation then, if you really need all the power of a symbolic package, buy either Maple or Mathematica, my preference going to Maple. If you want a system that is already specialized for number theoretic computations, then buy Magma. In any case, as a complement to this package, obtain Pari.

Where to obtain these packages.

You can order Maple at the following address: Waterloo Maple Software, 160 Columbia St. W., Waterloo, Ontario, Canada N2L 3L3, phone (519) 747-2373, fax (519) 747-5284, e-mail `wmsi@daisy.waterloo.edu`. Maple has been ported to many different machines and it is highly probable that it has been ported to the machine that you want. There is also a system named Mathcad that uses some parts of Maple for its symbolic manipulations; Mathcad runs under Microsoft Windows and is published by MathSoft Inc., 201 Broadway, Cambridge, Massachussets, USA, 02139 Phone: (617) 577-1017.

You can order Mathematica from Wolfram Research, Inc. at the following address: Wolfram Research, 100 Trade Center Drive, Champaign, IL 61820, phone 800-441-Math, fax 217-398-0747, e-mail `info@wri.com`. Mathematica

has also been ported to quite a number of machines, and in addition you can use a friendly "front-end" like the Macintosh II linked to a more powerful computer (including supercomputers) which will do the actual computations.

Macsyma exists in two flavors : the commercial versions (Macsyma, ALJABR, ParaMacs) are licensed from MIT, the non-commercial versions (Vaxima, Maxima, and DOE-Macsyma) officially come from the American Department of Energy (DOE). All these versions are derived from the Macsyma developed by the Mathlab Group at MIT. The commercial version runs on PC 386, Symbolics computers, VMS machines and most Unix workstations; the address to order it is: Macsyma Inc., 20 Academy Street, Suite 201, Arlington MA 02174-6436, phone (617) 646-4550 or 1-800-MACSYMA (free from the U.S.), fax (617) 646-3161, e-mail `info-macsyma@macsyma.com`. Vaxima is available from the Energy Science and Technology Software Center (ESTSC), P.O. Box 1020, Oak Ridge, Tennessee 37831, phone (615) 576-2606. Maxima is a Common Lisp version maintained by William Schelter (e-mail `wfs@math.utexas.edu`) at Texas University. Although it is a non-commercial version, one must get a license from the Energy Science and Technology Software Center (see above) to use it. For more information, get the file `README.MAXIMA` by anonymous ftp on `rascal.ics.utexas.edu`. ParaMacs, is available from Leo Harten, Paradigm Associates, Inc., 29 Putnam Avenue, Suite 6, Cambridge, MA 02139, phone (617) 492-6079, fax (617) 876-8186, e-mail `lph@paradigm.com`. ALJABR is available from Fort Pond Research, 15 Fort Pond Road, Acton, MA 01720, phone 508-263-9692, e-mail `aljabr@fpr.com`. It runs on Macintosh, Sun and SGI computers.

There are many distributors of Reduce, depending on the machine and version of Lisp that is used. The main one is Herbert Melenk, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Heilbronner Str. 10, D 1000 Berlin 31, Germany, phone 30-89604-195, fax 30-89604-125, e-mail `melenk@sc.zib-berlin.de`. You will get detailed informations if you send an electronic message with `send info-package` as subject to `reduce-netlib@rand.org`.

Axiom on IBM RS/6000 is distributed by NAG: contact the Numerical Algorithms Group Ltd., Wilkinson House, Jordan Hill Rd., Oxford, UK OX2 8DR, phone (0)-865-511245, e-mail `nagttt@vax.oxford.ac.uk`. A Sparc version is also available.

Derive is available from Soft Warehouse, Inc., 3615 Harding Avenue, Suite 505, Honolulu, Hawaii 96816, USA, phone (808) 734-5801, fax (808) 735-1105.

You can obtain Ubasic by anonymous ftp at `shape.mps.ohio-state.edu` or `wuarchive.wustl.edu`. Or you can write directly to Kida at the following address: Prof. Yuji Kida, Department of Mathematics, Rikkyo University, Nishi-Ikebukuro 3, Tokyo 171, JAPAN, e-mail `kida@rkmath.rikkyo.ac.jp`.

The Calculus Calculator (CC) is developed by David Meredith, Department of Mathematics, San Francisco State University, 1600 Holloway Avenue, San Francisco, CA 94132, phone (415) 338-2199. Version 3 (CC3) is published

with a 200 page manual by Prentice Hall, phone (201) 767-5937. Version 4 (CC4) is available by anonymous ftp from `wuarchive.wustl.edu`.

You can order Magma from The Secretary, Computational Algebra Group, Pure Mathematics, University of Sydney, NSW 2006, Australia, phone (2) 692-3338, fax (2) 692-4534, e-mail `magma@maths.su.oz.au`. It runs on Sun, HP, Apollo, VAX/VMS, Convex and various IBM machines.

GAP is available free of charge through ftp from Aachen: the ordinary mail address is Lehrstuhl D für Mathematik, RWTH Aachen, Templergraben 64, D-5100 Aachen, Germany. For technical questions, contact Martin Schoenert (e-mail `martin@math.rwth-aachen.de`), and for more general questions, contact Prof. Joachim Neubüser (e-mail `neubueser@math.rwth-aachen.de`).

There are two versions of Kant: Kant V1 is written in Ansi-Fortran 77, while Kant V2 is built on the Magma Platform and written in Ansi-C. These two versions are available from the KANT Group: e-mail to `pohst@math.tu-berlin.de` or `daberkow@math.tu-berlin.de`. You can get the system by anonymous ftp from `ftp.math.tu-berlin.de`, directory `/pub/algebra/Kant`. Note that Kant V2 is now also part of the Magma package.

You can obtain Simath by anonymous ftp from `ftp.math.uni-sb.de`.

Numbers is developed by Ivo Düntsch, Moorlandstr. 59, W-4500 Osnabrück, phone (541) 189-106, fax (541) 969-2470, e-mail `duentsch@dosuni1.bitnet`. You can get the system by anonymous ftp from `dione.rz.uni-osnabrueck.de`.

You can obtain Gmp (as well as all software from the Free Software Foundation) by anonymous ftp on `prep.ai.mit.edu`.

The three multi-precision systems named Calc can all be obtained by anonymous ftp: the GNU calculator (written and maintained by Dave Gillespie, e-mail `daveg@csvax.cs.caltech.edu`, 256-80 Caltech, Pasadena, CA 91125) from `csvax.cs.caltech.edu`, the calculator of Hans-J. Boehm from `arisia.xerox.com` and the calculator of Timothy C. Frenz (5361 Amalfi Drive, Clay, NY 13041) from the site `wuarchive.wustl.edu`.

Finally, you can obtain Pari by anonymous ftp from the sites `megrez.ceremab.u-bordeaux.fr`, `ftp.inria.fr` and `math.ucla.edu`.

Internet addresses and numbers for ftp

`arisia.xerox.com`	13.1.64.94	Boehm-Calc
`csvax.cs.caltech.edu`	131.215.131.131	GNU Calc
`dione.rz.uni-osnabrueck.de`	131.173.128.15	Numbers
`ftp.math.tu-berlin.de`	130.149.12.72	Kant
`ftp.math.uni-sb.de`	134.96.32.23	Simath
`math.ucla.edu`	128.97.4.254	Pari
`megrez.ceremab.u-bordeaux.fr`	147.210.16.17	Pari
`prep.ai.mit.edu`	18.71.0.38	Gmp
`rascal.ics.utexas.edu`	128.83.138.20	Maxima
`shape.mps.ohio-state.edu`	128.146.110.30	Ubasic
`wuarchive.wustl.edu`	128.252.135.4	Most packages

Appendix B

Some Useful Tables

In this appendix, we give five short tables which may be useful as basic data on which to work in algebraic number fields and on elliptic curves. The first two tables deal with quadratic fields and can be found in many places.

The third and fourth table give the corresponding tables for complex and totally real cubic fields respectively, and have been produced by M. Olivier using the method explained in Section 6.4.1 and the KANT package (see Appendix A).

The fifth table is a short table of elliptic curves extracted from [LN476] and [Cre].

I give here a list of references to the main tables that I am aware of. Not included are tables which have been superseded, and also papers containing only a few of the smallest number fields.

For quadratic fields see [Bue1] and [Ten-Wil].

For cubic fields see [Enn-Tur1], [Enn-Tur2], [Gras], [Ang], [Sha-Wil] and [Ten-Wil].

For quartic fields see [Ford3], [Buc-Ford] and [BFP].

For quintic fields see [Diaz] and [SPD].

For sextic fields see [Oli3], [Oli4], [Oli5] and [Oli6].

Finally, for an extensive table of elliptic curves see Cremona's book [Cre].

B.1 Table of Class Numbers of Complex Quadratic Fields

Recall that the group of units of complex quadratic fields is equal to ± 1 except when the discriminant is equal to -3 or -4 in which case it is equal to the group of sixth or fourth roots of unity respectively.

The following table list triples $(d, h(d), H(-d))$ where d is negative and congruent to 0 or 1 modulo 4, $h(d)$ is the class number of the quadratic order of discriminant d, and $H(-d)$ is the Hurwitz class number of discriminant d (see Definition 5.3.6). Note that $h(d) = H(-d)$ if and only if d is a fundamental discriminant, that $H(-d)$ has a denominator equal to 2 (resp. 3) if and only if d is of the form $-4f^2$ (resp. $-3f^2$) and otherwise is an integer.

(−3,1,1/3)	(−4,1,1/2)	(−7,1,1)	(−8,1,1)
(−11,1,1)	(−12,1,4/3)	(−15,2,2)	(−16,1,3/2)
(−19,1,1)	(−20,2,2)	(−23,3,3)	(−24,2,2)
(−27,1,4/3)	(−28,1,2)	(−31,3,3)	(−32,2,3)
(−35,2,2)	(−36,2,5/2)	(−39,4,4)	(−40,2,2)
(−43,1,1)	(−44,3,4)	(−47,5,5)	(−48,2,10/3)
(−51,2,2)	(−52,2,2)	(−55,4,4)	(−56,4,4)
(−59,3,3)	(−60,2,4)	(−63,4,5)	(−64,2,7/2)
(−67,1,1)	(−68,4,4)	(−71,7,7)	(−72,2,3)
(−75,2,7/3)	(−76,3,4)	(−79,5,5)	(−80,4,6)
(−83,3,3)	(−84,4,4)	(−87,6,6)	(−88,2,2)
(−91,2,2)	(−92,3,6)	(−95,8,8)	(−96,4,6)
(−99,2,3)	(−100,2,5/2)	(−103,5,5)	(−104,6,6)
(−107,3,3)	(−108,3,16/3)	(−111,8,8)	(−112,2,4)
(−115,2,2)	(−116,6,6)	(−119,10,10)	(−120,4,4)
(−123,2,2)	(−124,3,6)	(−127,5,5)	(−128,4,7)
(−131,5,5)	(−132,4,4)	(−135,6,8)	(−136,4,4)
(−139,3,3)	(−140,6,8)	(−143,10,10)	(−144,4,15/2)
(−147,2,7/3)	(−148,2,2)	(−151,7,7)	(−152,6,6)
(−155,4,4)	(−156,4,8)	(−159,10,10)	(−160,4,6)
(−163,1,1)	(−164,8,8)	(−167,11,11)	(−168,4,4)
(−171,4,5)	(−172,3,4)	(−175,6,7)	(−176,6,10)
(−179,5,5)	(−180,4,6)	(−183,8,8)	(−184,4,4)
(−187,2,2)	(−188,5,10)	(−191,13,13)	(−192,4,22/3)
(−195,4,4)	(−196,4,9/2)	(−199,9,9)	(−200,6,7)
(−203,4,4)	(−204,6,8)	(−207,6,9)	(−208,4,6)
(−211,3,3)	(−212,6,6)	(−215,14,14)	(−216,6,8)
(−219,4,4)	(−220,4,8)	(−223,7,7)	(−224,8,12)
(−227,5,5)	(−228,4,4)	(−231,12,12)	(−232,2,2)
(−235,2,2)	(−236,9,12)	(−239,15,15)	(−240,4,8)
(−243,3,13/3)	(−244,6,6)	(−247,6,6)	(−248,8,8)
(−251,7,7)	(−252,4,10)	(−255,12,12)	(−256,4,15/2)
(−259,4,4)	(−260,8,8)	(−263,13,13)	(−264,8,8)
(−267,2,2)	(−268,3,4)	(−271,11,11)	(−272,8,12)
(−275,4,5)	(−276,8,8)	(−279,12,15)	(−280,4,4)
(−283,3,3)	(−284,7,14)	(−287,14,14)	(−288,4,9)
(−291,4,4)	(−292,4,4)	(−295,8,8)	(−296,10,10)
(−299,8,8)	(−300,6,28/3)	(−303,10,10)	(−304,6,10)
(−307,3,3)	(−308,8,8)	(−311,19,19)	(−312,4,4)
(−315,4,6)	(−316,5,10)	(−319,10,10)	(−320,8,14)
(−323,4,4)	(−324,6,17/2)	(−327,12,12)	(−328,4,4)
(−331,3,3)	(−332,9,12)	(−335,18,18)	(−336,8,12)
(−339,6,6)	(−340,4,4)	(−343,7,8)	(−344,10,10)
(−347,5,5)	(−348,6,12)	(−351,12,16)	(−352,4,6)

(−355,4,4)	(−356,12,12)	(−359,19,19)	(−360,8,10)
(−363,4,13/3)	(−364,6,8)	(−367,9,9)	(−368,6,12)
(−371,8,8)	(−372,4,4)	(−375,10,12)	(−376,8,8)
(−379,3,3)	(−380,8,16)	(−383,17,17)	(−384,8,14)
(−387,4,5)	(−388,4,4)	(−391,14,14)	(−392,8,9)
(−395,8,8)	(−396,6,12)	(−399,16,16)	(−400,4,15/2)
(−403,2,2)	(−404,14,14)	(−407,16,16)	(−408,4,4)
(−411,6,6)	(−412,5,10)	(−415,10,10)	(−416,12,18)
(−419,9,9)	(−420,8,8)	(−423,10,15)	(−424,6,6)
(−427,2,2)	(−428,9,12)	(−431,21,21)	(−432,6,40/3)
(−435,4,4)	(−436,6,6)	(−439,15,15)	(−440,12,12)
(−443,5,5)	(−444,8,16)	(−447,14,14)	(−448,4,8)
(−451,6,6)	(−452,8,8)	(−455,20,20)	(−456,8,8)
(−459,6,8)	(−460,6,8)	(−463,7,7)	(−464,12,18)
(−467,7,7)	(−468,8,10)	(−471,16,16)	(−472,6,6)
(−475,4,5)	(−476,10,20)	(−479,25,25)	(−480,8,12)
(−483,4,4)	(−484,6,13/2)	(−487,7,7)	(−488,10,10)
(−491,9,9)	(−492,6,8)	(−495,16,20)	(−496,6,12)
(−499,3,3)	(−500,10,12)	(−503,21,21)	(−504,8,12)

B.2 Table of Class Numbers and Units of Real Quadratic Fields

In the following table of real quadratic fields K we list the following data from left to right: the discriminant $d = d(K)$, the class number $h = h(K)$, the regulator $R = R(K)$, the norm of the fundamental unit and finally the fundamental unit itself given as a pair of coordinates (a, b) on the canonical integral basis $(1, \omega)$ where $\omega = (1 + \sqrt{d})/2$ if $d \equiv 1 \pmod 4$, $\omega = \sqrt{d}/2$ if $d \equiv 0 \pmod 4$.

d	h	R	$\mathcal{N}(\epsilon)$	ϵ
5	1	0.4812	−1	(0,1)
8	1	0.8814	−1	(1,1)
12	1	1.317	1	(2,1)
13	1	1.195	−1	(1,1)
17	1	2.095	−1	(3,2)
21	1	1.567	1	(2,1)
24	1	2.292	1	(5,2)
28	1	2.769	1	(8,3)
29	1	1.647	−1	(2,1)
33	1	3.828	1	(19,8)
37	1	2.492	−1	(5,2)

40	2	1.818	−1	(3,1)
41	1	4.159	−1	(27,10)
44	1	2.993	1	(10,3)
53	1	1.966	−1	(3,1)
56	1	3.400	1	(15,4)
57	1	5.710	1	(131,40)
60	2	2.063	1	(4,1)
61	1	3.664	−1	(17,5)
65	2	2.776	−1	(7,2)
69	1	3.217	1	(11,3)
73	1	7.667	−1	(943,250)
76	1	5.829	1	(170,39)
77	1	2.185	1	(4,1)
85	2	2.209	−1	(4,1)
88	1	5.976	1	(197,42)
89	1	6.908	−1	(447,106)
92	1	3.871	1	(24,5)
93	1	3.366	1	(13,3)
97	1	9.324	−1	(5035,1138)
101	1	2.998	−1	(9,2)
104	2	2.312	−1	(5,1)
105	2	4.407	1	(37,8)
109	1	5.565	−1	(118,25)
113	1	7.347	−1	(703,146)
120	2	3.089	1	(11,2)
124	1	8.020	1	(1520,273)
129	1	10.43	1	(15371,2968)
133	1	5.153	1	(79,15)
136	2	4.248	1	(35,6)
137	1	8.157	−1	(1595,298)
140	2	2.478	1	(6,1)
141	1	5.247	1	(87,16)
145	4	3.180	−1	(11,2)
149	1	4.111	−1	(28,5)
152	1	4.304	1	(37,6)
156	2	3.912	1	(25,4)
157	1	5.361	−1	(98,17)
161	1	10.07	1	(10847,1856)
165	2	2.559	1	(6,1)
168	2	3.257	1	(13,2)
172	1	8.849	1	(3482,531)
173	1	2.571	−1	(6,1)
177	1	11.73	1	(57731,9384)
181	1	7.174	−1	(604,97)
184	1	10.79	1	(24335,3588)

185	2	4.913	−1	(63,10)
188	1	4.564	1	(48,7)
193	1	15.08	−1	(1637147,253970)
197	1	3.333	−1	(13,2)
201	1	13.85	1	(478763,72664)
204	2	4.605	1	(50,7)
205	2	3.761	1	(20,3)
209	1	11.44	1	(43331,6440)
213	1	4.290	1	(34,5)
217	1	15.86	1	(3583111,521904)
220	2	5.182	1	(89,12)
221	2	2.704	1	(7,1)
229	3	2.712	−1	(7,1)
232	2	5.288	−1	(99,13)
233	1	10.74	−1	(21639,3034)
236	1	6.966	1	(530,69)
237	1	4.344	1	(36,5)
241	1	18.77	−1	(66436843,9148450)
248	1	4.836	1	(63,8)
249	1	16.66	1	(8011739,1084152)
253	1	7.529	1	(872,117)
257	3	3.467	−1	(15,2)
264	2	4.867	1	(65,8)
265	2	9.405	−1	(5699,746)
268	1	11.49	1	(48842,5967)
269	1	5.100	−1	(77,10)
273	2	7.282	1	(683,88)
277	1	7.868	−1	(1228,157)
280	2	6.219	1	(251,30)
281	1	14.57	−1	(1000087,126890)
284	1	8.848	1	(3480,413)
285	2	2.830	1	(8,1)
293	1	2.837	−1	(8,1)
296	2	4.454	−1	(43,5)
301	1	10.03	1	(10717,1311)
305	2	6.886	1	(461,56)
309	1	8.526	1	(2379,287)
312	2	4.663	1	(53,6)
313	1	19.35	−1	(119691683,14341370)
316	3	5.075	1	(80,9)
317	1	4.489	−1	(42,5)
321	3	6.064	1	(203,24)
328	4	2.893	−1	(9,1)
329	1	15.37	1	(2245399,262032)
332	1	5.100	1	(82,9)

337	1	21.43	−1	(960491695,110671282)
341	1	5.624	1	(131,15)
344	1	9.943	1	(10405,1122)
345	2	9.512	1	(6397,728)
348	2	4.025	1	(28,3)
349	1	9.821	−1	(8717,986)
353	1	11.87	−1	(67471,7586)
357	2	2.942	1	(9,1)
364	2	8.055	1	(1574,165)
365	2	2.947	−1	(9,1)
373	1	9.234	−1	(4853,530)
376	1	15.27	1	(2143295,221064)
377	2	6.144	1	(221,24)
380	2	4.357	1	(39,4)
381	1	7.616	1	(963,104)
385	2	12.16	1	(90947,9768)
389	1	7.849	−1	(1217,130)
393	1	18.35	1	(44094699,4684888)
397	1	8.145	−1	(1637,173)
401	5	3.690	−1	(19,2)
408	2	5.308	1	(101,10)
409	1	26.13	−1	(106387620283,11068353370)
412	1	13.03	1	(227528,22419)
413	1	4.111	1	(29,3)
417	1	18.96	1	(81144379,8356536)
421	1	13.01	−1	(211627,21685)
424	2	8.988	−1	(4005,389)
428	1	7.562	1	(962,93)
429	2	4.977	1	(69,7)
433	1	23.39	−1	(6883177307,694966754)
437	1	3.042	1	(10,1)
440	2	3.737	1	(21,2)
444	2	6.380	1	(295,28)
445	4	3.047	−1	(10,1)
449	1	19.75	−1	(180529627,17883410)
453	1	5.004	1	(71,7)
456	2	7.626	1	(1025,96)
457	1	25.50	−1	(56325840235,5528222698)
460	2	7.720	1	(1126,105)
461	1	5.900	−1	(174,17)
465	2	10.37	1	(15135,1472)
469	3	4.174	1	(31,3)
472	1	13.33	1	(306917,28254)
473	3	5.159	1	(83,8)
476	2	5.481	1	(120,11)

481	2	14.47	−1	(920179,87922)
485	2	3.785	−1	(21,2)
488	2	3.093	−1	(11,1)
489	1	23.44	1	(7249279379,686701192)
492	2	5.497	1	(122,11)
493	2	4.710	−1	(53,5)
497	1	14.69	1	(1147975,107824)

B.3 Table of Class Numbers and Units of Complex Cubic Fields

Any number field can be defined as $K = \mathbb{Q}[\alpha]$ where α is a primitive algebraic integer (see Section 10.5.2), and we will denote by $A(X)$ the minimal monic polynomial of α. We will choose A so that the index $f = [\mathbb{Z}_K : \mathbb{Z}[\alpha]]$ is as small as possible and with small coefficients (hence A will not always be the pseudo-canonical polynomial given by Algorithm 4.4.12). The choice of the particular polynomials A which we will give is therefore not at all canonical.

Let now K be a cubic field. Since we have chosen α primitive, there exists an integral basis of the form $(1, \alpha, \beta)$. Furthermore any cubic field has at least one real embedding hence the set of roots of unity is always equal to ± 1. On the other hand complex cubic fields have unit rank equal to 1, while real cubic fields have unit rank equal to 2. Since the norm of -1 is equal to -1, there is no such thing as the sign of the norm of fundamental units.

The following is a table of the first hundred complex cubic fields. For each field K we give the following data from left to right: the discriminant $d = d(K)$, the index $f = [\mathbb{Z}_K : \mathbb{Z}[\alpha]]$, the polynomial A, the third element β of an integral basis $(1, \alpha, \beta)$, the class number $h = h(K)$, the regulator $R = R(K)$ and the fundamental unit ϵ expressed on the integral basis (for example $(2, 3, 1)$ means $2 + 3\alpha + \beta$). Since the signature of K is equal to $(1, 1)$, the Galois group of the Galois closure of K is always equal to the symmetric group S_3.

d	f	A	β	h	R	ϵ
-23	1	$X^3 + X^2 - 1$	α^2	1	0.2812	(0,1,1)
-31	1	$X^3 - X^2 - 1$	α^2	1	0.3822	(0,1,0)
-44	1	$X^3 - X^2 - X - 1$	α^2	1	0.6094	(0,1,0)
-59	1	$X^3 + 2X - 1$	α^2	1	0.7910	(2,0,1)
-76	1	$X^3 - 2X - 2$	α^2	1	1.019	(1,1,0)
-83	1	$X^3 - X^2 + X - 2$	α^2	1	1.041	(1,0,1)
-87	1	$X^3 + X^2 + 2X - 1$	α^2	1	0.9348	(2,1,1)
-104	1	$X^3 - X - 2$	α^2	1	1.576	(1,1,1)
-107	1	$X^3 - X^2 + 3X - 2$	α^2	1	1.256	(3,0,1)
-108	1	$X^3 - 2$	α^2	1	1.347	(1,1,1)

-116	1	$X^3 - X^2 - 2$	α^2	1	1.718	(1,1,1)
-135	1	$X^3 + 3X - 1$	α^2	1	1.133	(3,0,1)
-139	1	$X^3 + X^2 + X - 2$	α^2	1	1.664	(3,2,1)
-140	1	$X^3 + 2X - 2$	α^2	1	1.474	(3,1,1)
-152	1	$X^3 - X^2 - 2X - 2$	α^2	1	2.131	(-1,-1,-1)
-172	1	$X^3 + X^2 - X - 3$	α^2	1	1.882	(-2,-2,-1)
-175	1	$X^3 - X^2 + 2X - 3$	α^2	1	1.289	(2,0,1)
-199	1	$X^3 - X^2 + 4X - 1$	α^2	1	1.337	(4,-1,1)
-200	1	$X^3 + X^2 + 2X - 2$	α^2	1	2.604	(9,5,3)
-204	1	$X^3 - X^2 + X - 3$	α^2	1	2.355	(4,1,2)
-211	1	$X^3 - 2X - 3$	α^2	1	2.238	(-2,-2,-1)
-212	1	$X^3 - X^2 + 4X - 2$	α^2	1	2.713	(-15,2,-4)
-216	1	$X^3 + 3X - 2$	α^2	1	3.024	(-17,-3,-5)
-231	1	$X^3 + X^2 - 3$	α^2	1	1.745	(2,2,1)
-239	1	$X^3 - X - 3$	α^2	1	2.097	(2,2,1)
-243	1	$X^3 - 3$	α^2	1	2.525	(4,3,2)
-244	1	$X^3 + X^2 - 4X - 6$	α^2	1	3.303	(5,6,2)
-247	1	$X^3 + X - 3$	α^2	1	1.545	(2,1,1)
-255	1	$X^3 - X^2 - 3$	α^2	1	1.993	(-2,-1,-1)
-268	1	$X^3 + X^2 - 3X - 5$	α^2	1	2.521	(3,3,1)
-283	1	$X^3 + 4X - 1$	α^2	2	1.401	(4,0,1)
-300	1	$X^3 - X^2 - 3X - 3$	α^2	1	3.149	(2,3,2)
-307	1	$X^3 + X^2 + 3X - 2$	α^2	1	2.958	(-15,-6,-4)
-324	1	$X^3 - 3X - 4$	α^2	1	4.048	(-9,-11,-5)
-327	1	$X^3 - X^2 - 2X - 3$	α^2	1	2.199	(1,1,1)
-331	1	$X^3 - X^2 + 3X - 4$	α^2	2	1.503	(3,0,1)
-335	1	$X^3 + X^2 + 4X - 1$	α^2	1	1.456	(4,1,1)
-339	1	$X^3 + X^2 - X - 4$	α^2	1	3.546	(11,10,4)
-351	1	$X^3 + 3X - 3$	α^2	1	1.702	(-4,-1,-1)
-356	2	$X^3 - X^2 + 4X - 8$	$(\alpha + \alpha^2)/2$	1	3.755	(-25,2,-10)
-364	1	$X^3 + 4X - 2$	α^2	1	2.936	(17,2,4)
-367	1	$X^3 + X^2 + 2X - 3$	α^2	1	1.856	(4,2,1)
-379	1	$X^3 - X^2 + X - 4$	α^2	1	3.273	(9,3,4)
-411	1	$X^3 - X^2 + 5X - 2$	α^2	1	4.029	(57,-7,12)
-419	1	$X^3 - 4X - 5$	α^2	1	3.345	(-4,-5,-2)
-424	2	$X^3 - 2X - 8$	$\alpha^2/2$	1	4.859	(31,21,18)
-431	2	$X^3 - X - 8$	$(\alpha + \alpha^2)/2$	1	6.155	(133,42,72)
-436	1	$X^3 + X - 4$	α^2	1	4.948	(-61,-29,-21)
-439	1	$X^3 + X^2 - 2X - 5$	α^2	1	2.430	(3,3,1)
-440	2	$X^3 + 2X - 8$	$\alpha^2/2$	1	4.534	(-43,-15,-18)
-451	1	$X^3 + X^2 - 5X - 8$	α^2	1	3.576	(-7,-7,-2)
-459	1	$X^3 - 6X - 7$	α^2	1	3.669	(-5,-6,-2)
-460	1	$X^3 - X^2 + 5X - 3$	α^2	1	3.671	(38,-3,8)
-472	1	$X^3 - 5X - 6$	α^2	1	5.380	(29,35,13)
-484	1	$X^3 + X^2 + 4X - 2$	α^2	1	5.303	(171,53,37)
-491	1	$X^3 + X^2 + X - 4$	α^2	2	1.891	(3,2,1)
-492	1	$X^3 + X^2 + 3X - 3$	α^2	1	4.421	(59,24,14)
-499	1	$X^3 + 4X - 3$	α^2	1	3.874	(-40,-6,-9)
-503	2	$X^3 - X^2 - 2X - 8$	$(\alpha + \alpha^2)/2$	1	7.027	(-211,-56,-146)
-515	1	$X^3 - X^2 - X - 4$	α^2	1	3.646	(-7,-5,-4)
-516	2	$X^3 - X^2 - 4X - 8$	$(\alpha + \alpha^2)/2$	1	6.385	(-81,-35,-63)
-519	1	$X^3 + X^2 - 4X - 7$	α^2	1	2.681	(3,3,1)
-524	1	$X^3 - X^2 + 3X - 5$	α^2	1	3.422	(18,2,5)
-527	1	$X^3 + 5X - 1$	α^2	1	1.617	(5,0,1)

-543	1	$X^3 - X^2 + 2X - 5$	α^2	1	3.013	(-9,-2,-3)
-547	1	$X^3 - X^2 - 3X - 4$	α^2	1	4.367	(9,10,6)
-563	1	$X^3 - X^2 + 5X - 4$	α^2	2	1.737	(5,0,1)
-567	1	$X^3 - 3X - 5$	α^2	1	2.464	(-2,-2,-1)
-588	1	$X^3 + X^2 + 5X - 1$	α^2	3	1.654	(5,1,1)
-620	1	$X^3 - X^2 - 5X - 5$	α^2	1	3.553	(3,4,2)
-628	2	$X^3 + X^2 - 3X - 11$	$(1+\alpha^2)/2$	1	6.494	(-138,-123,-74)
-643	1	$X^3 - 2X - 5$	α^2	2	2.359	(2,2,1)
-648	2	$X^3 - 3X - 10$	$(\alpha+\alpha^2)/2$	3	2.234	(2,1,1)
-652	1	$X^3 - 8X - 10$	α^2	1	4.320	(-11,0,1)
-655	1	$X^3 + X^2 - 5$	α^2	1	2.906	(-7,-5,-2)
-671	1	$X^3 - X - 5$	α^2	1	2.345	(-3,-2,-1)
-675	1	$X^3 - 5$	α^2	1	4.812	(-41,-24,-14)
-676	2	$X^3 + X^2 - 4X - 12$	$(\alpha+\alpha^2)/2$	3	2.186	(2,1,1)
-679	1	$X^3 + X - 5$	α^2	1	3.443	(13,6,4)
-680	1	$X^3 + X^2 - 6X - 10$	α^2	1	6.071	(-79,-77,-21)
-687	1	$X^3 + X^2 + 4X - 3$	α^2	1	3.455	(-25,-8,-5)
-695	1	$X^3 - X^2 - 5$	α^2	1	2.151	(2,1,1)
-696	1	$X^3 + X^2 - 2X - 6$	α^2	1	7.810	(-673,-589,-207)
-707	1	$X^3 + 2X - 5$	α^2	1	4.187	(34,12,9)
-716	1	$X^3 - 4X - 6$	α^2	1	6.405	(-95,-101,-40)
-728	1	$X^3 - X^2 + 6X - 2$	α^2	1	6.052	(-433,49,-75)
-731	1	$X^3 + X^2 + 3X - 4$	α^2	2	2.013	(-5,-2,-1)
-743	1	$X^3 + 5X - 3$	α^2	1	4.556	(-85,-9,-16)
-744	1	$X^3 - X^2 - 6X - 6$	α^2	1	8.294	(-347,-451,-193)
-748	1	$X^3 + X^2 + X - 5$	α^2	1	4.532	(-43,-25,-11)
-751	1	$X^3 - X^2 + 6X - 1$	α^2	2	1.768	(6,-1,1)
-755	1	$X^3 + X^2 + 5X - 2$	α^2	1	4.904	(121,30,22)
-756	2	$X^3 + 9X - 2$	$(\alpha+\alpha^2)/2$	1	7.107	(1208,-104,267)
-759	1	$X^3 - X^2 + 6X - 3$	α^2	1	3.137	(23,-2,4)
-771	1	$X^3 - X^2 + 3X - 6$	α^2	1	6.140	(-251,-36,-65)
-780	1	$X^3 - X^2 - X - 5$	α^2	1	6.159	(94,59,44)
-804	1	$X^3 - X^2 + 4X - 6$	α^2	1	8.571	(-3499,-270,-784)
-808	1	$X^3 - X^2 + 2X - 6$	α^2	1	7.625	(-875,-201,-259)
-812	1	$X^3 - X^2 - 7X - 7$	α^2	1	3.844	(4,5,2)
-815	1	$X^3 - 7X - 9$	α^2	1	5.064	(20,22,7)

B.4 Table of Class Numbers and Units of Totally Real Cubic Fields

The following is a table of the first hundred totally real cubic fields. We give the following data from left to right: the discriminant $d(K)$, the index $[\mathbb{Z}_K : \mathbb{Z}[\alpha]]$, the polynomial $A(X)$, the third element β of an integral basis $(1, \alpha, \beta)$, the class number $h(K)$, the regulator $R(K)$ and a pair of fundamental units ϵ_1 and ϵ_2 expressed on the integral basis $(1, \alpha, \beta)$. The Galois group of the Galois closure of K is equal to S_3 except for the fields whose discriminant is marked with an asterisk, which are cyclic cubic fields, i.e. with Galois group equal to C_3.

d	f	A	β	h	R	ϵ_1	ϵ_2
49*	1	X^3+X^2-2X-1	α^2	1	0.5255	(-1,1,1)	(2,0,-1)
81*	1	X^3-3X-1	α^2	1	0.8493	(2,1,-1)	(0,-1,0)
148	1	X^3+X^2-3X-1	α^2	1	1.662	(0,1,0)	(2,0,-1)
169*	1	X^3-X^2-4X-1	α^2	1	1.365	(2,2,-1)	(0,-1,0)
229	1	X^3-4X-1	α^2	1	2.355	(0,1,0)	(2,1,0)
257	1	X^3-5X-3	α^2	1	1.975	(4,1,-1)	(5,1,-1)
316	1	X^3+X^2-4X-2	α^2	1	3.913	(-3,1,1)	(-5,1,1)
321	1	X^3+X^2-4X-1	α^2	1	2.569	(0,-1,0)	(-1,2,1)
361*	1	X^3+X^2-6X-7	α^2	1	1.952	(4,1,-1)	(5,0,-1)
404	1	X^3-X^2-5X-1	α^2	1	3.760	(0,-1,0)	(1,-1,-1)
469	1	X^3+X^2-5X-4	α^2	1	3.853	(-1,-1,0)	(-1,2,1)
473	1	X^3-5X-1	α^2	1	2.843	(0,-1,0)	(-2,-1,0)
564	1	X^3+X^2-5X-3	α^2	1	5.403	(-2,1,0)	(-1,-1,1)
568	1	X^3-X^2-6X-2	α^2	1	6.087	(-5,-1,1)	(-7,-4,2)
621	1	X^3-6X-3	α^2	1	5.400	(-2,-1,0)	(1,2,0)
697	1	X^3-X^2-8X-5	α^2	1	2.712	(6,2,-1)	(7,2,-1)
733	1	X^3+X^2-7X-8	α^2	1	5.309	(1,1,0)	(-5,-2,0)
756	1	X^3-6X-2	α^2	1	5.692	(5,0,-1)	(11,1,-2)
761	1	X^3-X^2-6X-1	α^2	1	3.526	(0,1,0)	(2,1,0)
785	1	X^3+X^2-6X-5	α^2	1	4.098	(1,1,0)	(-4,1,1)
788	1	X^3-X^2-7X-3	α^2	1	5.987	(2,1,0)	(-1,-2,0)
837	1	X^3-6X-1	α^2	1	6.801	(0,-1,0)	(-3,-6,-2)
892	1	$X^3+X^2-8X-10$	α^2	1	8.323	(3,1,-1)	(1,3,1)
940	1	X^3-7X-4	α^2	1	8.908	(-11,-2,2)	(-3,1,1)
961*	2	$X^3+X^2-10X-8$	$(\alpha^2+\alpha)/2$	1	12.20	(-1,2,2)	(3,4,-2)
985	1	X^3+X^2-6X-1	α^2	1	3.724	(0,1,0)	(-2,1,0)
993	1	X^3+X^2-6X-3	α^2	1	5.555	(5,-1,-1)	(5,0,-1)
1016	1	X^3+X^2-6X-2	α^2	1	10.13	(7,-1,-1)	(-11,-1,1)
1076	1	X^3-8X-6	α^2	1	6.932	(1,1,0)	(-7,-3,0)
1101	1	$X^3+X^2-9X-12$	α^2	1	9.184	(5,2,-1)	(-7,-4,2)
1129	1	X^3-7X-3	α^2	1	6.728	(-8,0,1)	(1,2,-1)
1229	1	X^3+X^2-7X-6	α^2	1	8.232	(-1,-1,0)	(11,15,4)
1257	1	X^3+X^2-8X-9	α^2	1	6.197	(-1,-1,0)	(2,-2,-1)
1300	1	$X^3-10X-10$	α^2	1	6.550	(-1,-1,0)	(-1,2,1)
1304	2	$X^3-X^2-11X-1$	$(\alpha^2+1)/2$	1	11.93	(0,-1,0)	(-5,14,10)
1345	1	X^3-7X-1	α^2	1	4.923	(0,1,0)	(2,2,-1)
1369*	1	$X^3-X^2-12X-11$	α^2	1	3.126	(6,3,-1)	(9,2,-1)
1373	1	X^3-8X-5	α^2	1	9.423	(-6,0,1)	(-13,-2,2)
1384	1	$X^3+X^2-10X-14$	α^2	1	10.38	(-3,-2,0)	(-5,1,1)
1396	1	X^3+X^2-7X-5	α^2	1	8.146	(-8,0,1)	(-9,1,1)
1425	1	X^3-X^2-8X-3	α^2	1	6.676	(-2,-1,0)	(1,2,-1)
1436	1	$X^3-11X-12$	α^2	1	12.70	(5,2,0)	(-11,-6,2)
1489	1	$X^3+X^2-12X-19$	α^2	1	3.361	(10,1,-1)	(11,1,-1)
1492	1	X^3-X^2-9X-5	α^2	1	7.646	(-2,-1,0)	(-1,-1,1)
1509	1	X^3+X^2-7X-4	α^2	1	11.30	(3,1,0)	(-3,-6,-1)
1524	1	X^3+X^2-7X-1	α^2	1	10.45	(0,1,0)	(-6,-11,6)
1556	1	$X^3+X^2-9X-11$	α^2	1	8.376	(8,0,-1)	(19,0,-2)
1573	1	X^3+X^2-7X-2	α^2	1	8.445	(-3,-1,0)	(1,4,1)
1593	1	X^3-9X-7	α^2	1	6.331	(1,1,0)	(5,2,0)
1620	1	$X^3-12X-14$	α^2	1	10.17	(9,1,-1)	(5,5,1)
1708	1	X^3-X^2-8X-2	α^2	1	12.87	(7,1,-1)	(-29,-9,5)
1765	1	$X^3+X^2-11X-16$	α^2	1	9.445	(-3,-1,0)	(-7,-6,-1)

1772	2	$X^3 - 14X - 12$	$\alpha^2/2$	1	15.37	(-1,-1,0)	(-23,-36,-18)
1825	1	$X^3 + X^2 - 8X - 7$	α^2	1	4.488	(1,1,0)	(3,1,0)
1849*	2	$X^3 - X^2 - 14X - 8$	$(\alpha^2 + \alpha)/2$	1	18.92	(-9,2,0)	(-17,-4,2)
1901	1	$X^3 - X^2 - 9X - 4$	α^2	1	10.66	(-1,-2,0)	(-5,0,1)
1929	1	$X^3 + X^2 - 10X - 13$	α^2	1	8.218	(3,1,0)	(5,5,1)
1937	1	$X^3 - X^2 - 8X - 1$	α^2	1	6.542	(0,-1,0)	(-3,1,1)
1940	1	$X^3 - 8X - 2$	α^2	1	11.09	(3,-1,0)	(39,1,-5)
1944	1	$X^3 - 9X - 6$	α^2	1	15.60	(1,3,-1)	(-1,0,2)
1957	1	$X^3 + X^2 - 9X - 10$	α^2	2	4.551	(1,1,0)	(3,1,0)
2021	1	$X^3 - 8X - 1$	α^2	1	11.52	(0,-1,0)	(-1,-28,-10)
2024	1	$X^3 - X^2 - 10X - 6$	α^2	1	15.77	(5,6,-2)	(-11,-9,3)
2057	1	$X^3 - 11X - 11$	α^2	1	6.782	(1,1,0)	(-1,-3,-1)
2089	2	$X^3 - 13X - 4$	$(\alpha^2 + \alpha)/2$	1	20.76	(-1,-4,2)	(-15,4,0)
2101	1	$X^3 - X^2 - 11X - 8$	α^2	1	8.543	(-1,-1,0)	(15,2,-2)
2177	1	$X^3 + X^2 - 8X - 5$	α^2	1	7.518	(-3,-1,0)	(17,-1,-2)
2213	1	$X^3 - X^2 - 13X - 12$	α^2	1	12.68	(-1,-1,0)	(-1,9,4)
2228	1	$X^3 - 14X - 18$	α^2	1	11.09	(-7,-3,1)	(-41,-16,6)
2233	1	$X^3 + X^2 - 8X - 1$	α^2	1	5.523	(0,1,0)	(-1,3,1)
2241	1	$X^3 - 9X - 5$	α^2	1	8.264	(-4,-2,1)	(-2,-3,1)
2292	2	$X^3 + X^2 - 13X - 1$	$(\alpha^2 + 1)/2$	1	14.36	(0,1,0)	(-4,36,17)
2296	1	$X^3 - X^2 - 14X - 14$	α^2	1	14.27	(13,3,-1)	(-5,-4,0)
2300	1	$X^3 + X^2 - 8X - 2$	α^2	1	18.12	(5,-2,0)	(73,-7,-9)
2349	1	$X^3 - 12X - 13$	α^2	1	11.92	(-4,-2,1)	(15,4,-2)
2429	1	$X^3 - X^2 - 15X - 16$	α^2	1	13.28	(-11,-2,1)	(85,16,-7)
2505	1	$X^3 - X^2 - 10X - 5$	α^2	1	10.68	(-2,-3,1)	(7,6,-2)
2557	1	$X^3 - X^2 - 9X - 2$	α^2	1	10.72	(-1,2,1)	(1,4,-1)
2589	2	$X^3 + X^2 - 14X - 12$	$(\alpha^2 + \alpha)/2$	1	16.29	(-5,-1,1)	(31,38,-20)
2597	1	$X^3 + X^2 - 9X - 8$	α^2	3	4.796	(1,1,0)	(-3,-1,0)
2636	1	$X^3 - X^2 - 16X - 18$	α^2	1	18.38	(-5,-2,0)	(25,13,-3)
2673	1	$X^3 - 9X - 3$	α^2	1	7.760	(10,0,-1)	(-8,0,1)
2677	1	$X^3 - 10X - 7$	α^2	1	11.16	(-12,0,1)	(2,2,-1)
2700	1	$X^3 - 15X - 20$	α^2	1	20.37	(1,-1,-1)	(-59,-22,8)
2708	1	$X^3 - X^2 - 11X - 7$	α^2	1	12.95	(6,7,-2)	(9,6,-2)
2713	1	$X^3 - 13X - 15$	α^2	1	12.34	(-13,-2,1)	(-17,-4,2)
2777	1	$X^3 + X^2 - 14X - 23$	α^2	2	3.949	(-2,-1,0)	(-3,-1,0)
2804	1	$X^3 - X^2 - 9X - 1$	α^2	1	15.24	(0,-1,0)	(10,56,21)
2808	1	$X^3 - 9X - 2$	α^2	1	20.31	(-1,-9,3)	(-1,-4,2)
2836	1	$X^3 + X^2 - 9X - 7$	α^2	1	9.692	(10,0,-1)	(-17,0,2)
2857	1	$X^3 + X^2 - 10X - 11$	α^2	1	4.870	(-1,-1,0)	(-3,-1,0)
2917	1	$X^3 + X^2 - 13X - 20$	α^2	1	11.93	(3,1,0)	(13,6,-1)
2920	2	$X^3 + X^2 - 16X - 20$	$(\alpha^2 + \alpha)/2$	1	17.94	(-9,-8,4)	(-4,-3,1)
2941	1	$X^3 - X^2 - 17X - 20$	α^2	1	13.72	(3,2,0)	(-17,-4,1)
2981	1	$X^3 + X^2 - 11X - 14$	α^2	1	14.63	(3,1,0)	(15,10,-1)
2993	1	$X^3 + X^2 - 12X - 17$	α^2	1	7.514	(-3,-1,0)	(3,2,0)
3021	1	$X^3 + X^2 - 9X - 6$	α^2	1	17.40	(-5,-4,2)	(5,9,2)
3028	1	$X^3 - 10X - 6$	α^2	1	20.35	(-1,-1,1)	(5,13,4)
3124	2	$X^3 - 16X - 12$	$\alpha^2/2$	1	19.56	(-5,-1,1)	(115,121,-68)
3132	2	$X^3 - 18X - 20$	$\alpha^2/2$	1	22.49	(7,2,0)	(7,7,2)

B.5 Table of Elliptic Curves

In the table below we give a table of all modular elliptic curves defined over $\mathbb{Q}$ with conductor N less than or equal to 44 (up to isomorphism). Recall that according to the Taniyama-Weil Conjecture 7.3.8, all elliptic curves defined over $\mathbb{Q}$ are modular.

To every elliptic curve is attached quite a large set of invariants. We refer to [Cre] for details and a complete table. In the following table, we only give the minimal Weierstraß equation of the curve, its rank and its torsion subgroup. The rank is always equal to 0 except in the two cases $N = 37$ (curve A1) and $N = 43$ for which it is equal to 1, and in these two cases a generator of the group $E(\mathbb{Q})$ is the point with coordinates $(0,0)$. The canonical height of this point, computed using Algorithms 7.5.6 and 7.5.7 is equal to $0.0255557041\ldots$ for $N = 37$ and to $0.0314082535\ldots$ for $N = 43$.

The Kodaira types and the constants c_p can be found by using Tate's Algorithm 7.5.1. The coefficients a_p of the L-series can be computed using Algorithm 7.4.12 or simply by adding Legendre symbols if p is small. The periods can be computed using Algorithm 7.4.7. In the limit of the present table the Tate-Shafarevitch group Ш is always trivial.

We follow the notations of [Cre]. We give from left to right: the conductor N of the curve E, an identifying label of the curve among those having the same conductor. This label is of the form letter-number. The letter (A or B) denotes the isogeny class, and the number is the ordinal number of the curve in its isogeny class. Curves numbered 1 are the strong Weil curves (see [Sil]). The next 5 columns contain the coefficients a_1, a_2, a_3, a_4 and a_6. The last two columns contain the rank r and the torsion subgroup T of $E(\mathbb{Q})$ expressed as t if $T \simeq \mathbb{Z}/t\mathbb{Z}$ and as $t_1 \times t_2$ if $T \simeq \mathbb{Z}/t_1\mathbb{Z} \times \mathbb{Z}/t_2\mathbb{Z}$.

N		a_1	a_2	a_3	a_4	a_6	r	T
11	A1	0	−1	1	−10	−20	0	5
11	A2	0	−1	1	−7820	−263580	0	1
11	A3	0	−1	1	0	0	0	5
14	A1	1	0	1	4	−6	0	6
14	A2	1	0	1	−36	−70	0	6
14	A3	1	0	1	−171	−874	0	2
14	A4	1	0	1	−1	0	0	6
14	A5	1	0	1	−2731	−55146	0	2
14	A6	1	0	1	−11	12	0	6
15	A1	1	1	1	−10	−10	0	2×4
15	A2	1	1	1	−135	−660	0	2×2
15	A3	1	1	1	−5	2	0	2×4
15	A4	1	1	1	35	−28	0	8
15	A5	1	1	1	−2160	−39540	0	2

15	A6	1	1	1	−110	−880	0	2
15	A7	1	1	1	−80	242	0	4
15	A8	1	1	1	0	0	0	4
17	A1	1	−1	1	−1	−14	0	4
17	A2	1	−1	1	−6	−4	0	2 × 2
17	A3	1	−1	1	−91	−310	0	2
17	A4	1	−1	1	−1	0	0	4
19	A1	0	1	1	−9	−15	0	3
19	A2	0	1	1	−769	−8470	0	1
19	A3	0	1	1	1	0	0	3
20	A1	0	1	0	4	4	0	6
20	A2	0	1	0	−1	0	0	6
20	A3	0	1	0	−36	−140	0	2
20	A4	0	1	0	−41	−116	0	2
21	A1	1	0	0	−4	−1	0	2 × 4
21	A2	1	0	0	−49	−136	0	2 × 2
21	A3	1	0	0	−39	90	0	8
21	A4	1	0	0	1	0	0	4
21	A5	1	0	0	−784	−8515	0	2
21	A6	1	0	0	−34	−217	0	2
24	A1	0	−1	0	−4	4	0	2 × 4
24	A2	0	−1	0	−24	−36	0	2 × 2
24	A3	0	−1	0	−64	220	0	4
24	A4	0	−1	0	1	0	0	4
24	A5	0	−1	0	−384	−2772	0	2
24	A6	0	−1	0	16	−180	0	2
26	A1	1	0	1	−5	−8	0	3
26	A2	1	0	1	−460	−3830	0	1
26	A3	1	0	1	0	0	0	3
26	B1	1	−1	1	−3	3	0	7
26	B2	1	−1	1	−213	−1257	0	1
27	A1	0	0	1	0	−7	0	3
27	A2	0	0	1	−270	−1708	0	1
27	A3	0	0	1	0	0	0	3
27	A4	0	0	1	−30	63	0	3
30	A1	1	0	1	1	2	0	6
30	A2	1	0	1	−19	26	0	2 × 6
30	A3	1	0	1	−14	−64	0	2
30	A4	1	0	1	−69	−194	0	6
30	A5	1	0	1	−289	1862	0	6
30	A6	1	0	1	−334	−2368	0	2 × 2
30	A7	1	0	1	−5334	−150368	0	2
30	A8	1	0	1	−454	−544	0	2
32	A1	0	0	0	4	0	0	4
32	A2	0	0	0	−1	0	0	2 × 2

32	A3	0	0	0	−11	−14	0	2
32	A4	0	0	0	−11	14	0	4
33	A1	1	1	0	−11	0	0	2×2
33	A2	1	1	0	−6	−9	0	2
33	A3	1	1	0	−146	621	0	4
33	A4	1	1	0	44	55	0	2
34	A1	1	0	0	−3	1	0	6
34	A2	1	0	0	−43	105	0	6
34	A3	1	0	0	−103	−411	0	2
34	A4	1	0	0	−113	−329	0	2
35	A1	0	1	1	9	1	0	3
35	A2	0	1	1	−131	−650	0	1
35	A3	0	1	1	−1	0	0	3
36	A1	0	0	0	0	1	0	6
36	A2	0	0	0	−15	22	0	6
36	A3	0	0	0	0	−27	0	2
36	A4	0	0	0	−135	−594	0	2
37	A1	0	0	1	−1	0	1	1
37	B1	0	1	1	−23	−50	0	3
37	B2	0	1	1	−1873	−31833	0	1
37	B3	0	1	1	−3	1	0	3
38	A1	1	0	1	9	90	0	3
38	A2	1	0	1	−86	−2456	0	1
38	A3	1	0	1	−16	22	0	3
38	B1	1	1	1	0	1	0	5
38	B2	1	1	1	−70	−279	0	1
39	A1	1	1	0	−4	−5	0	2×2
39	A2	1	1	0	−69	−252	0	2
39	A3	1	1	0	−19	22	0	4
39	A4	1	1	0	1	0	0	2
40	A1	0	0	0	−7	−6	0	2×2
40	A2	0	0	0	−107	−426	0	2
40	A3	0	0	0	−2	1	0	4
40	A4	0	0	0	13	−34	0	4
42	A1	1	1	1	−4	5	0	8
42	A2	1	1	1	−84	261	0	2×4
42	A3	1	1	1	−104	101	0	2×2
42	A4	1	1	1	−1344	18405	0	4
42	A5	1	1	1	−914	−10915	0	2
42	A6	1	1	1	386	1277	0	2
43	A1	0	1	1	0	0	1	1
44	A1	0	1	0	3	−1	0	3
44	A2	0	1	0	−77	−289	0	1

Bibliography

Essential Introductory Books.

[**Bo-Sh**] Z.I. Borevitch and I.R. Shafarevitch, *Number Theory*, Academic Press, New York, 1966.

A classic must which gives a fairly advanced introduction to algebraic number theory, with applications for example to Fermat's last theorem. Contains numerous exercises.

[**GCL**] K. Geddes, S. Czapor and G. Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1992.

This book contains a very detailed description of the basic algorithms used for handling fundamental mathematical objects such as polynomials, power series, rational functions, as well as more sophisticated algorithms such as polynomial factorization, Gröbner bases computation and symbolic integration. The algorithms are those which have been implemented in the Maple system (see Appendix A). This is required reading for anyone wanting to understand the inner workings of a computer algebra system.

[**H-W**] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, (5-th ed.), Oxford University Press, Oxford, 1979.

This is another classic must for a beginning introduction to number theory. The presentation is very clear and simple, and the book contains all basic essential material. Avoid reading parts like the "elementary" proof of the prime number theorem. Proofs based on complex function theory, while requiring deeper concepts, are much more enlightening.

[**Ire-Ros**] K. Ireland and M. Rosen, *A Classical Introduction to Modern Number Theory*, (2nd ed.), Graduate texts in Math. **84**, Springer-Verlag, New York, 1982.

A remarkable introductory book on the more analytic and computational parts of algebraic number theory, with numerous concrete examples and exercises. This book can be read profitably jointly with the present book for a deeper understanding of several subjects such as Gauss and Jacobi sums and related identities (used in Chapter 9), quadratic and cyclotomic fields (Chapters 5 and 9), zeta functions of varieties (Chapter 7), etc .... A must.

[**Knu1**] D.E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, (2nd ed.), Addison-Wesley, Reading, Mass., 1973.

This is the first volume of the "bible" of computer science. Although not specifically targeted to number theory, this volume introduces a large number of fundamental concepts and techniques (mathematical or otherwise) which are of constant use to anyone implementing algorithms. The style of writing is crystal clear, and I have copied the style of presentation of algorithms from Knuth. A must.

[**Knu2**] D.E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, (2nd ed.), Addison-Wesley, Reading, Mass., 1981.

This is the second volume of the "bible" of computer science. Essentially all the contents of chapter 4 of Knuth's book is basic to computational number theory, and as stated in the preface, some parts of chapters 1 and 3 of the present book have been merely adapted from Knuth. The section on factoring and primality testing is of course outdated. The book contains also a huge number of fascinating exercises, with solutions. An absolute must.

[**Knu3**] D.E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973.

This is the third volume of the "bible" of computer science. The description of searching and sorting methods (in particular heapsort and quicksort) as well as hashing techniques can be used for number-theoretic applications.

One can find at the URL

`http://www-cs-faculty.stanford.edu/~knuth/index.html`

nearly 350 pages of corrections and additions to [Knu1], [Knu2] and [Knu3], absolutely necessary for those having the older editions of Knuth's books. This has been incorporated in a new 3 volume set which came out in 1996.

[**Lang1**] S. Lang, *Algebra*, (2nd ed.), Addison-Wesley, Reading, Mass., 1984.

This book is quite abstract in nature and in fact contains little concrete examples. On the other hand one can find the statements and proofs of most of the basic algebraic results needed in number theory.

[**Mar**] D.A. Marcus, *Number Fields*, Springer-Verlag, New York, 1977.

An excellent textbook on algebraic number theory with numerous very concrete examples, not far from the spirit of this book, although much less algorithmic in nature.

[**Rie**] H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Birkhäuser, Boston, 1985.

An excellent elementary text on prime number theory and algorithms for primality testing and factoring. As in the present book the algorithms are ready to implement, and in fact implementations of many of them are given in Pascal. The subject matter of the algorithmic part overlaps in a large part with chapters 8 to 10 of this book.

[**Sam**] P. Samuel, *Théorie algébrique des nombres*, Hermann, Paris, 1971.

Another excellent textbook on algebraic number theory. Gives the basic proofs and results in a very nice and concise manner.

[**Ser**] J.-P. Serre, *A Course in Arithmetic*, Springer-Verlag, New York, 1973.

A very nice little book which contains an introduction to some basic number-theoretic objects such as $\mathbb{Z}/n\mathbb{Z}$, finite fields, quadratic forms, modular forms, etc .... A must, although further reading is necessary in almost all cases. The original was published in French in 1970.

Other Books and Volumes.

[**AHU**] A. Aho, J. Hopcroft and J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.

This book discusses many issues related to basic computer algorithms and their complexity. In particular, it discusses in detail the notion of NP-complete problems, and has chapters on integer and polynomial arithmetic, on the LUP decomposition of matrices and on the Fast Fourier transform.

[**Bac-Sha**] E. Bach and J. Shallit, *Algorithmic Number Theory, Vol. 1: Efficient Algorithms*, MIT Press, Cambridge, Mass, 1996.

Studies in detail the complexity of number-theoretic algorithms.

[**Bor-Bor**] J. Borwein and P. Borwein, *Pi and the AGM*, Canadian Math. Soc. Series, John Wiley and Sons, New York, 1987.

A marvelous book containing a wealth of formulas in the style of Ramanujan, including formulas coming from complex multiplication for computing π to great accuracy.

[**Bue**] D. Buell, *Binary Quadratic Forms: Classical Theory and Modern Computations*, Springer-Verlag, New York, 1990.

A nice and easy to read book on the theory of binary quadratic forms, which expands on some of the subjects treated in Chapter 5.

[**Cas**] J. Cassels, *Lectures on elliptic curves*, Cambridge Univ. Press, 1991.

An excellent small introductory book to the subject of elliptic curves containing a wealth of deeper subjects not so easily accessible otherwise. The viewpoint is different from Silverman's, and hence is a highly recommended complementary reading.

[**Cas-Frö**] J. Cassels and A. Fröhlich, *Algebraic number theory*, Academic Press, London and New York, 1967.

This book has been one of the main reference books for a generation of algebraic number theorists and is still the standard book to read before more sophisticated books like Shimura's.

[**Cohn**] H. Cohn, *A Classical Introduction to Algebraic Numbers and Class Fields*, Universitext, Springer-Verlag, New York, 1978.

A highly recommended concrete introduction to algebraic number theory and class field theory, with a large number of detailed examples.

[**Con-Slo**] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups*, Grundlehren der math. Wiss. **290**, Springer-Verlag, New York, 1988.

The bible on lattices and sphere packings. Everything you ever wanted to know and much more, including a large number of tables. An irreplaceable tool for research in the Geometry of Numbers.

[**Cox**] D. Cox, *Primes of the Form $x^2 + ny^2$. Fermat, Class Field Theory and Complex Multiplication*, John Wiley and Sons, New York, 1989.

This is an excellent book on class field theory and complex multiplication. It is written in a very concrete manner with many examples and exercises, and I recommend it highly.

[**Cre**] J. Cremona, *Algorithms for Modular Elliptic Curves*, Cambridge Univ. Press, 1992.

An extension of [LN476] to conductors less than 1000, and much more information. Also many algorithms related to elliptic curves are listed, most of which are not given in this book. A must on the subject.

[**Dah-Bjö**] G. Dahlquist and A. Björk (translated by N. Anderson), *Numerical Methods*, Prentice Hall, Englewood Cliffs, N.J., 1974.

A basic reference book on numerical algorithms, especially for linear algebra.

[**Del-Fad**] B.N. Delone and D.K. Fadeev, *The Theory of Irrationalities of the Third Degree*, Trans. Math. Mon. **10**, A.M.S., Providence, R.I., 1964.

Although quite old, this book contains a wealth of theoretical and algorithmic information on cubic fields.

[**Gol-Van**] G. Golub and C. Van Loan, *Matrix Computations*, (2nd ed.), Johns Hopkins Univ. Press, Baltimore and London, 1989.

An excellent comprehensive introduction to basic techniques of numerical analysis used in linear algebra.

[**Hus**] D. Husemoller, *Elliptic Curves*, Graduate texts in Math. **111**, Springer-Verlag, New York, 1987.

Simpler than Silverman's book, this gives a good introduction to elliptic curves.

[**Kap**] I. Kaplansky, *Commutative Rings*, Allyn and Bacon, Boston, 1970.

A very nicely written little book on abstract algebra.

[**Kob**] N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, Graduate texts in Math. **97**, Springer-Verlag, New York, 1984.

This nice book gives the necessary tools for obtaining the complete solution of the congruent number problem modulo a weak form of the Birch-Swinnerton Dyer conjecture. In passing, a lot of very concrete material on elliptic curves and modular forms is covered.

[**Lang2**] S. Lang, *Algebraic Number Theory*, Addison-Wesley, Reading, Mass., 1970.

An advanced abstract introduction to the subject.

[**Lang3**] S. Lang, *Elliptic Functions*, Addison Wesley, Reading, Mass., 1973.

A nice introductory book on elliptic functions and elliptic curves.

[**Lang4**] S. Lang, *Introduction to Modular Forms*, Springer-Verlag, Berlin, Heidelberg, New York, 1976.

A nice introductory book on modular forms.

[**LN476**] B. Birch and W. Kuyk (eds.), *Modular Forms in one Variable IV*, LN in Math. **476**, Springer-Verlag, Berlin, Heidelberg, 1975.

A fundamental book of tables and algorithms on elliptic curves, containing in particular a detailed description of all elliptic curves of conductor less than or equal to 200. A must on the subject.

[**MCC**] H.W. Lenstra and R. Tijdeman (eds.), *Computational Methods in Number Theory*, Math. Centre tracts **154/155**, Math. Centrum Amsterdam, 1982.

A very nice two volume collection on computational number theory, covering many different topics.

[**Nau-Qui**] P. Naudin and C. Quitté, *Algorithmique Algébrique*, Masson, Paris, 1992.

A very nice and leisurely introduction to computational algebra (in French) with many detailed algorithms and a complete chapter devoted to the use of the Fast Fourier Transform in computer algebra.

[**Ogg**] A. Ogg, *Modular Forms and Dirichlet Series*, Benjamin, 1969.

A nice little introductory book on modular forms, containing in particular a detailed proof of Weil's Theorem 7.3.7.

[**PPWZ**] A. Pethő, M. Pohst, H. Williams and H.G. Zimmer (eds.), *Computational Number Theory*, Walter de Gruyter, 1991.

Similar to [MCC] but very up to date and more oriented towards algebraic number theory. Contains very important contributions which are referenced separately here.

[**Poh**] M. Pohst (ed.), *Algorithmic Methods in Algebra and Number Theory*, Academic Press, 1987.

A special volume of the Journal of Symbolic Computation devoted to computational number theory, and containing a number of important individual contributions which are referenced separately here.

[**Poh-Zas**] M. Pohst and H. Zassenhaus, *Algorithmic Algebraic Number Theory*, Cambridge Univ. Press, 1989.

The reference book on algorithmic algebraic number theory. Contains detailed descriptions of numerous algorithms for solving the fundamental tasks of algebraic number theory in the general number field case. The notation is sometimes heavy, and direct computer implementation of the algorithms is not always easy, but the wealth of information is considerable. A must for further reading on the subject.

[**Poh5**] M. Pohst, *Computational Algebraic Number Theory*, DMV Seminar **21**, Birkhäuser, Boston, 1993.

Writeup of a course given by the author in 1990. This can be considered as an update to parts of [Poh-Zas].

[**PFTV**] W. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes in C*, (2nd ed.), Cambridge University Press, Cambridge, 1988.

The algorithms presented in this book are essentially unrelated to number theory, but this is a basic reference book for implementing algorithms in numerical analysis, and in particular for number theory, polynomial root finding and linear algebra over $\mathbb{R}$. A must for implementing numerical analysis-related algorithms.

[**Sha**] H. Williams (ed.), Math. Comp **48(January)** (1987).

A special volume of Mathematics of Computation dedicated to D. Shanks. Contains a large number of important individual contributions.

[**Shi**] G. Shimura, *Introduction to the Arithmetic Theory of Automorphic Functions*, Iwanami Shoten and Princeton Univ. Press, Princeton, 1971.

This book is one of the great classics of advanced number theory, in particular about class fields, elliptic curves and modular forms. It contains a great wealth of information, and even though it is quite old, it is still essentially up to date and still a basic reference book. Beware however that the mathematical sophistication is high. A must for people wanting to know more about class fields, complex multiplication and modular forms at a high level.

[**Sil**] J. Silverman, *The Arithmetic of Elliptic Curves*, Graduate texts in Math. **106**, Springer-Verlag, New York, 1986.

This excellent book has now become *the* reference book on elliptic curves, and a large part is of very advanced level. It is excellently written, contains numerous exercises and is a great pleasure to study. A must for further study of elliptic curves.

[**Sil3**] J. Silverman, *Advanced Topics in the Arithmetic of Elliptic Curves*, Graduate texts in Math. **151**, Springer-Verlag, New York, 1994.

The long awaited sequel to [Sil].

[**Was**] L. Washington, *Introduction to Cyclotomic Fields*, Graduate Texts in Math. **83**, Springer-Verlag, New York, 1982.

An excellent advanced introduction to algebraic number theory, with many concrete examples.

[**W-W**] E. Whittaker and G. Watson, *A Course of Modern Analysis*, (4th ed.), Cambridge Univ. Press, 1927.

Still the reference book on practical use of complex analysis. The chapters on elliptic functions and theta functions are of special interest to number theorists.

[**Zag**] D. Zagier, *The Analytic Theory of Modular Forms*, in preparation.

A thorough introduction to the analytic theory of modular forms, including a number of advanced topics. Very clear exposition. A must on the subject (when it comes out).

[**Zim**] H. Zimmer, *Computational Problems, Methods and Results in Algebraic Number Theory*, LN in Math. **262**, Springer-Verlag, Berlin, Heidelberg, 1972.

A very thorough list of commented bibliographic references on computational number theory prior to 1971.

Papers and other references

[**Adl**] L. Adleman, *Factoring numbers using singular integers*, Proc. 18th Annual ACM Symp. on Theory of Computing (1991), 64–71.

[**Adl-Hua**] L. Adleman and M. Huang, *Primality testing and Abelian varieties over finite fields*, LN in Math **1512**, Springer-Verlag, Berlin, Heidelberg, 1992.

[**APR**] L. Adleman, C. Pomerance and R. Rumely, *On distinguishing prime numbers from composite numbers*, Ann. of Math. **117** (1983), 173–206.

[**AGP**] R. Alford, A. Granville and C. Pomerance, *There are infinitely many Carmichael numbers*, Ann. of Math. **139** (1994), 703–722.

[**Ang**] I. Angell, *A table of complex cubic fields*, Bull. London Math. Soc. **5** (1973), 37–38.

[**Arn**] F. Arnault, *The Rabin-Miller primality test: composite numbers which pass it*, Math. Comp. **64** (1995), 335–361.

[**ARW**] S. Arno, M. Robinson and F. Wheeler, *Imaginary quadratic fields with small odd class number* (to appear).

[**Atk1**] O. Atkin, *Composition of binary quadratic forms*, manuscript (1990).

[**Atk2**] O. Atkin, *The number of points on an elliptic curve modulo a prime*, manuscript (1991).

[**Atk-Mor**] O. Atkin and F. Morain, *Elliptic curves and primality proving*, Math. Comp. **61** (1993), 29–68.

[**Ayo**] R. Ayoub, *An Introduction to the Analytic Theory of Numbers*, Mathematical surveys **10**, A.M.S., 1963.

[**Bach**] E. Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990), 355–380.

[**Bar**] E. Bareiss, *Sylvester's identity and multistep integer-preserving Gaussian elimination*, Math. Comp. **22** (1968), 565–578.

[**BeMaOl**] A.-M. Bergé, J. Martinet and M. Olivier, *The computation of sextic fields with a quadratic subfield*, Math. Comp. **54** (1990), 869–884.

[**Ber**] E. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. **24** (1970), 713–735.

[**Bir-SwD**] B. Birch and H.P.F. Swinnerton-Dyer, *Notes on elliptic curves I*, J. Reine Angew. Math. **212** (1963), 7–25; *II*, ibid. **218** (1965), 79–108.

[**BFHT**] A. Borodin, R. Fagin, J. Hopcroft and M. Tompa, *Decreasing the nesting depth of expressions involving square roots*, J. Symb. Comp. **1** (1985), 169–188.

[**Bos**] W. Bosma, *Primality testing using elliptic curves*, Report 85-12, Math. Instituut, Univ. of Amsterdam (1985).

[**Bos-Hul**] W. Bosma and M.-P. van der Hulst, *Primality proving with cyclotomy*, thesis, Univ. of Amsterdam, 1990.

[**Bra**] G. Bradley, *Algorithms for Hermite and Smith normal form matrices and linear Diophantine equations*, Math. Comp. **25** (1971), 897–907.

[**Brau**] R. Brauer, *On the Zeta-function of algebraic number fields I*, Amer. J. Math. **69** (1947), 243–250; *II*, ibid. **72** (1950), 739–746.

[**Bre1**] R.P. Brent, *Some integer factorization algorithms using elliptic curves*, in Proc. 9th Australian Computer science conference (1985).

[**Bre2**] R.P. Brent, *An improved Monte-Carlo factorization algorithm*, BIT **20** (1980), 176–184.

[**Bre3**] R.P. Brent, *The first occurence of large gaps between successive primes*, Math. Comp. **27** (1973), 959–963.

[**BLSTW**] J. Brillhart, D.H. Lehmer, J. Selfridge, B. Tuckerman and S. Wagstaff, *Factorizations of $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$, up to high powers*, Contemporary Mathematics **22**, A.M.S., Providence, R.I., 1983.

[**Bri-Mor**] J. Brillhart and M. Morrison, *A method of factoring and the factorization of F_7*, Math. Comp. **29** (1975), 183–205.

[**BLS**] J. Brillhart, D.H. Lehmer and J. Selfridge, *New primality criteria and factorizations of $2^m \pm 1$*, Math. Comp. **29** (1975), 620–647.

[**BCS**] S. Brlek, P. Castéran and R. Strandh, *On addition schemes*, TAPSOFT 1991, LN in Comp. Sci. **494**, 1991, pp. 379–393.

[**deBru**] N. G. de Bruijn, *The asymptotic behavior of a function occurring in the theory of primes*, J. Indian Math. Soc. (N. S.) **15** (1951), 25–32.

[**Buc1**] J. Buchmann, *A generalization of Voronoi's unit algorithm I and II*, J. Number Theory **20** (1985), 177–209.

[**Buc2**] J. Buchmann, *On the computation of units and class numbers by a generalization of Lagrange's algorithm*, J. Number Theory **26** (1987), 8-30.

[**Buc3**] J. Buchmann, *On the period length of the generalized Lagrange algorithm*, J. Number Theory **26** (1987), 31–37.

[**Buc4**] J. Buchmann, *Zur Komplexität der Berechnung von Einheiten und Klassenzahlen algebraischer Zahlkörper*, Habilitationsschrift, University of Düsseldorf, 1988.

[**Buc-Dül**] J. Buchmann and S. Düllmann, *A probabilistic class group and regulator algorithm and its implementation*, in [PPWZ], 1991, pp. 53–72.

[**Buc-Ford**] J. Buchmann and D. Ford, *On the computation of totally real quartic fields of small discriminant*, Math. Comp. **52** (1989), 161-174.

[**BFP**] J. Buchmann, D. Ford and M. Pohst, *Enumeration of quartic fields of small discriminant*, Math. Comp. **61** (1993), 873–879.

[**Buc-Len**] J. Buchmann and H.W. Lenstra, *Computing maximal orders and factoring over* $\mathbb{Z}_p$, preprint.

[**Buc-Len2**] J. Buchmann and H.W. Lenstra, *Approximating rings of integers in number fields*, J. Th. des Nombres Bordeaux (Série 2) **6** (1994), 221–260.

[**Buc-Pet**] J. Buchmann and A. Pethő, *On the computation of independent units in number fields by Dirichlet's method*, Math. Comp. **52** (1989), 149–159.

[**Buc-Poh-Sch**] J. Buchmann, M. Pohst and J. Graf von Schmettow, *On the computation of unit groups and class groups of totally real quartic fields*, Math. Comp. **53** (1989), 387–397.

[**Buc-Thi-Wil**] J. Buchmann, C. Thiel and H. Williams, *Short representation of quadratic integers*, Computational Algebra and Number Theory, Mathematics and its Applications, Kluwer, Dordrecht, 1995, pp. 159–185.

[**Buc-Wil**] J. Buchmann and H. Williams, *On principal ideal testing in algebraic number fields*, J. Symb. Comp. **4** (1987), 11–19.

[**Buel**] D. Buell, *The expectation of success using a Monte-Carlo factoring method—some statistics on quadratic class numbers*, Math. Comp. **43** (1984), 313–327.

[**BGZ**] J. Buhler, B. Gross and D. Zagier, *On the conjecture of Birch and Swinnerton-Dyer for an elliptic curve of rank 3*, Math. Comp. **44** (1985), 473–481.

[**BLP**] J. Buhler, H. W. Lenstra and C. Pomerance, *Factoring integers with the number field sieve*, [Len-Len2], 1993, pp. 50–94.

[**But-McKay**] G. Butler and J. McKay, *The transitive groups of degree up to eleven*, Comm. in Algebra **11** (1983), 863–911.

[**CEP**] E.R. Canfield, P. Erdős and C. Pomerance, *On a problem of Oppenheim concerning "Factorisatio Numerorum"*, J. Number Theory **17** (1983), 1–28.

[**Can-Zas**] D. Cantor and H. Zassenhaus, *A new algorithm for factoring polynomials over finite fields*, Math. Comp. **36** (1981), 587–592.

[**Car**] H. Carayol, *Sur les représentations l-adiques associées aux formes modulaires de Hilbert*, Ann. Sci. E.N.S. **19** (1986), 409–468.

[**Chu**] D. and G. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factorization tests*, Adv. in Appl. Math. **7** (1986), 187–237.

[**Coa-Wil**] J. Coates and A. Wiles, *On the conjecture of Birch and Swinnerton-Dyer*, Invent. Math. **39** (1977), 223–251.

[**Coh1**] H. Cohen, *Variations sur un thème de Siegel et Hecke*, Acta Arith. **30** (1976), 63–93.

[**Coh2**] H. Cohen, *Formes modulaires à une et deux variables*, Thesis, Univ. de Bordeaux I, 1976.

[**Coh3**] P. Cohen, *On the coefficients of the transformation polynomials for the elliptic modular function*, Math. Proc. Cambridge Phil. Soc. **95** (1984), 389–402.

[**Coh-Diaz**] H. Cohen and F. Diaz y Diaz, *A polynomial reduction algorithm*, Sem. Th. Nombres Bordeaux (Série 2) **3** (1991), 351–360.

[**CohDiOl**] H. Cohen, F. Diaz y Diaz and M. Olivier, *Calculs de nombres de classes et de régulateurs de corps quadratiques en temps sous-exponentiel*, Séminaire de Théorie des Nombres Paris 1990–91 (1993), 35–46.

[**Coh-Len1**] H. Cohen and H.W. Lenstra, *Heuristics on class groups of number fields*, Number Theory, Noordwijkerhout 1983, LN in Math. **1068**, Springer-Verlag, 1984, pp. 33–62.

[**Coh-Len2**] H. Cohen and H.W. Lenstra, *Primality testing and Jacobi sums*, Math. Comp. **42** (1984), 297–330.

[**Coh-Len3**] H. Cohen and A.K. Lenstra, *Implementation of a new primality test*, Math. Comp. **48** (1987), 103–121.

[**Coh-Mar1**] H. Cohen and J. Martinet, *Class groups of number fields: numerical heuristics*, Math. Comp. **48** (1987), 123–137.

[**Coh-Mar2**] H. Cohen and J. Martinet, *Etude heuristique des groupes de classes des corps de nombres*, J. Reine Angew. Math. **404** (1990), 39–76.

[**Coh-Mar3**] H. Cohen and J. Martinet, *Heuristics on class groups: some good primes are not too good*, Math. Comp. **63** (1994), 329–334.

[**Col**] G. Collins, *The calculation of multivariate polynomial resultants*, JACM **18** (1971), 515–532.

[**Cop1**] D. Coppersmith, *Solving linear equations over* GF(2), RC **16997**, IBM Research, T.J. Watson research center (1991).

[**Cop2**] D. Coppersmith, *Solving homogeneous linear equations over* GF(2) *via block Wiedemann algorithm*, Math. Comp. **62** (1994), 333–350.

[**Del**] P. Deligne, *La conjecture de Weil I*, Publ. Math. IHES **43** (1974), 273–307.

[**Deu**] Deuring, *Die Klassenkörper der komplexen Multiplication*, Enzyklopädie der mathematischen Wissenschaften **12** (Book 10, Part II), Teubner, Stuttgart, 1958.

[**Diaz**] F. Diaz y Diaz, *A table of totally real quintic number fields*, Math. Comp. **56** (1991), 801–808 and S1–S12.

[**DKT**] P. Domich, R. Kannan and L. Trotter, *Hermite normal form computation using modulo determinant arithmetic*, Math. Oper. Research **12** (1987), 50–59.

[**Duk**] W. Duke, *Hyperbolic distribution functions and half-integral weight Maass forms*, Invent. Math. **92** (1988), 73–90.

[**Duv**] D. Duval, *Diverses questions relatives au calcul formel avec des nombres algébriques*, Thesis, Univ. of Grenoble, 1987.

[**Eic1**] Y. Eichenlaub, *Méthodes de calcul des groupes de Galois sur* $\mathbb{Q}$, Mémoire DEA, 1990.

[**Eic2**] M. Eichler, *On the class number of imaginary quadratic fields and the sums of divisors of natural numbers*, J. Indian Math. Soc. **19** (1955), 153–180.

[**Enn-Tur1**] V. Ennola and R. Turunen, *On totally real cubic fields*, Math. Comp. **44** (1985), 495–518.

[**Enn-Tur2**] V. Ennola and R. Turunen, *On cyclic cubic fields*, Math. Comp. **45** (1985), 585–589.

[**Fal**] G. Faltings, *Endlichkeitssätze für abelsche Varietäten über Zahlkörpern*, Invent. Math. **73** (1983), 349–366.

[**Fer1**] S. Fermigier, *Un exemple de courbe elliptique définie sur* $\mathbb{Q}$ *de rang* ≥ 19, C.R. Acad. Sci. Paris **315** (1992), 719–722.

[**Fer2**] S. Fermigier, in preparation.

[**Fin-Poh**] U. Fincke and M. Pohst, *Improved methods for calculating vectors of short length in a lattice, including a complexity analysis*, Math. Comp. **44** (1985), 463–471.

[**Ford1**] D. Ford, *On the computation of the maximal order in a Dedekind domain*, Thesis, Ohio State Univ., 1978.

[**Ford2**] D. Ford, *The construction of maximal orders over a Dedekind domain*, J. Symb. Comp. **4** (1987), 69–75.

[**Ford3**] D. Ford, *Enumeration of totally complex quartic fields of small discriminant*, in [PPWZ], 1991, pp. 129–138.

[**Fri**] E. Friedman, *Analytic formulas for the regulator of a number field*, Invent. math. **98** (1989), 599–622.

[**Gir**] K. Girstmair, *On invariant polynomials and their application in field theory*, Math. Comp. **48** (1987), 781–797.

[**Gol**] D. Goldfeld, *The class number of quadratic fields and the conjectures of Birch and Swinnerton-Dyer*, Ann. Sc. Norm. Super. Pisa **3** (1976), 623–663.

[**Gol-Kil**] S. Goldwasser and J. Kilian, *Almost all primes can be quickly certified*, Proc. 18th Annual ACM Symp. on Theory of Computing (1986), 316–329.

[**Gras**] M.-N. Gras, *Méthodes et algorithmes pour le calcul numérique du nombre de classes et des unités des extensions cubiques cycliques de* $\mathbb{Q}$, J. Reine Angew. Math. **277** (1975), 89–116.

[**Gro-Zag1**] B. Gross and D. Zagier, *On singular moduli*, J. Reine Angew. Math. **355** (1985), 191–220.

[**Gro-Zag2**] B. Gross and D. Zagier, *Heegner points and derivatives of L-series*, Invent. Math. **84** (1986), 225–320.

[**GKZ**] B. Gross, W. Kohnen and D. Zagier, *Heegner points and derivatives of L-series II*, Math. Ann. **278** (1987), 497–562.

[**Haf-McCur1**] J. Hafner and K. McCurley, *A rigorous subexponential algorithm for computation of class groups*, Journal American Math. Soc. **2** (1989), 837–850.

[**Haf-McCur2**] J. Hafner and K. McCurley, *Asymptotically fast triangularization of matrices over rings*, SIAM J. Comput. **20** (1991), 1068–1083.

[**Has**] H. Hasse, *Arithmetische Theorie der kubischen Zahlkörper auf klassenkörpertheoretischer Grundlage*, Math. Zeit. **31** (1930), 565–582; *Math. Abhandlungen*, Walter de Gruyter, 1975, pp. 423–440.

[**HJLS**] J. Hastad, B. Just, J.C. Lagarias and C.P. Schnorr, *Polynomial time algorithms for finding integer relations among real numbers*, Siam J. Comput. **18** (1989), 859–881.

[**Her**] O. Hermann, *Über die Berechnung der Fouriercoefficienten der Funktion $j(\tau)$*, J. Reine Angew. Math. **274/275** (1975), 187–195.

[**Hül**] A. Hülpke, in preparation.

[**Hun**] J. Hunter, *The minimum discriminants of quintic fields*, Proc. Glasgow Math. Ass. **3** (1957), 57–67.

[**Kal-Yui**] E. Kaltofen and N. Yui, *Explicit construction of the Hilbert class fields of imaginary quadratic fields by integer lattice reduction*, New York Number Theory Seminar 1989–1990, Springer-Verlag, 1991, pp. 150–202.

[**Kam**] S. Kamienny, *Torsion points on elliptic curves and q-coefficients of modular forms*, Invent. Math. **109** (1992), 221-229.

[**Kan-Bac**] R. Kannan and A. Bachem, *Polynomial algorithms for computing the Smith and Hermite normal form of an integer matrix*, Siam J. Comput. **8** (1979), 499-507.

[**Kol1**] V.A. Kolyvagin, *Finiteness of $E(\mathbb{Q})$ and $Ш(E/\mathbb{Q})$ for a subclass of Weil curves*, Izv. Akad. Nauk. SSSR **52** (1988), 522–540.

[**Kol2**] V.A. Kolyvagin, *Euler systems*, Progress in Math. **87**, Grothendieck Festschrift II, Birkhäuser, Boston, 1991, pp. 435–483.

[**LaM**] B. LaMacchia, *Basis reduction algorithms and subset sum problems*, Thesis, MIT Artificial Intelligence Lab., 1991.

[**LaM-Odl**] B. LaMacchia and A.M. Odlyzko, *Solving large sparse linear systems over finite fields*, Advances in cryptology: Crypto 90, A. Menezes and S. Vanstone (eds.), LN in Comp. Sci. **537**, Springer-Verlag, 1991, pp. 109–133.

[**Las**] M. Laska, *An algorithm for finding a minimal Weierstraß equation for an elliptic curve*, Math. Comp. **38** (1982), 257–260.

[**Leh1**] S. Lehman, *Factoring large integers*, Math. Comp. **28** (1974), 637–646.

[**Leh2**] D.H. Lehmer, *On Fermat's quotient, base two*, Math. Comp. **36** (1981), 289–290.

[**Len1**] H.W. Lenstra, *On the computation of regulators and class numbers of quadratic fields*, Lond. Math. Soc. Lect. Note Ser. **56** (1982), 123–150.

[**Len2**] H.W. Lenstra, *Divisors in residue classes*, Math. Comp. **42** (1984), 331-334.

[**Len3**] H.W. Lenstra, *Factoring integers with elliptic curves*, Ann. of Math. **126** (1987), 649–673.

[**Len4**] A.K. Lenstra, *Polynomial time algorithms for the factorization of polynomials*, dissertation, Univ. of Amsterdam, 1984.

[**Len-Len1**] A.K. Lenstra and H.W. Lenstra, *Algorithms in number theory*, Handbook of theoretical computer science, J. Van Leeuwen, A. Mayer, M. Nivat, M. Patterson and D. Perrin (eds.), Elsevier, Amsterdam, 1990, pp. 673–715.

[**Len-Len2**] A.K. Lenstra and H.W. Lenstra (eds.), *The development of the number field sieve*, LN in Math. **1554**, Springer-Verlag, Berlin, Heidelberg, New-York, 1993.

[**LLL**] A.K. Lenstra, H.W. Lenstra and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), 515–534.

[**LLMP**] A.K. Lenstra, H.W. Lenstra, M.S. Manasse and J.M. Pollard, *The Number Field Sieve*, in [Len-Len2], 1993, pp. 11–42.

[**Llo-Quer**] P. Llorente and J. Quer, *On the 3-Sylow subgroup of the class group of quadratic fields*, Math. Comp. **50** (1988), 321–333.

[**Mah**] K. Mahler, *On a class of non-linear functional equations connected with modular functions*, J. Austral. Math. Soc. **22A** (1976), 65–118.

[**Mart**] J. Martinet, *Méthodes géométriques dans la recherche des petits discriminants*, Progress in Math **59**, 1985, pp. 147–179.

[**Maz**] B. Mazur, *Rational isogenies of prime degree*, Invent. Math. **44** (1978), 129–162.

[**McCur**] K. McCurley, *Cryptographic key distribution and computation in class groups*, Proceedings of NATO ASI Number Theory and applications, Kluwer Academic Publishers, 1989, pp. 459–479.

[**Mer**] L. Merel, *Bornes pour la torsion des courbes elliptiques sur les corps de nombres*, Invent. Math. **124** (1996), 437–449.

[**Mes1**] J.-F. Mestre, *Construction d'une courbe elliptique de rang* ≥ 12, C.R. Acad. Sci. Paris **295** (1982), 643–644.

[**Mes2**] J.-F. Mestre, *Formules explicites et minorations de conducteurs de variétés algébriques*, Compositio Math. **58** (1986), 209–232.

[**Mes3**] J.-F. Mestre, *Courbes elliptiques de rang* ≥ 12 *sur* $\mathbb{Q}(t)$, C.R. Acad. Sci. Paris (1991), 171–174.

[**Mes4**] J.-F. Mestre, *Un exemple de courbe elliptique sur* $\mathbb{Q}$ *de rang* ≥ 15, C.R. Acad. Sci. Paris **314** (1992), 453–455.

[**Mes5**] J.-F. Mestre, private communication.

[**Mig**] M. Mignotte, *An inequality about factors of polynomials*, Math. Comp. **28** (1974), 1153–1157.

[**Mil**] G. Miller, *Riemann's hypothesis and tests for primality*, J. Comput. and System Sc. **13** (1976), 300–317.

[**Mol-Wil**] R. Mollin and H. Williams, *Computation of the class number of a real quadratic field*, Utilitas Math. **41** (1992), 259–308.

[**Mon-Nar**] J. Montes and E. Nart, *On a theorem of Ore*, Journal of Algebra **146** (1992), 318–339.

[**Mon1**] P. Montgomery, *Modular multiplication without trial division*, Math. Comp. **44** (1985), 519–521.

[**Mon2**] P. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Math. Comp. **48** (1987), 243–264.

[**Mor1**] F. Morain, *Résolution d'équations de petit degré modulo de grands nombres premiers*, Rapport de recherche INRIA **1085** (1989).

[**Mor2**] F. Morain, *Courbes elliptiques et tests de primalité*, Thesis, Univ. Claude Bernard, Lyon, 1990.

[**Mor-Nic**] F. Morain and J.-L. Nicolas, *On Cornacchia's algorithm for solving the Diophantine equation* $u^2 + dv^2 = m$ (to appear).

[**Nag**] K. Nagao, *An example of elliptic curve over* $\mathbb{Q}(T)$ *with rank* ≥ 13, Proc. Japan Acad. **70** (1994), 152–153.

[**Nag-Kou**] K. Nagao and T. Kouya, *An example of elliptic curve over* $\mathbb{Q}$ *with rank* ≥ 21, Proc. Japan Acad. **70** (1994), 104–105.

[**Nic**] J.-L. Nicolas, *Etre ou ne pas être un carré*, Dopo le Parole, (a collection of not always serious papers for A. K. Lenstra's doctorate), Amsterdam, 1984.

[**Odl**] A.M. Odlyzko, *Bounds for discriminants and related estimates for class numbers, regulators and zeros of zeta functions: a survey of recent results*, Sem. Th. des Nombres Bordeaux (Série 2) **2** (1991), 117–141.

[**Oes**] J. Oesterlé, *Nombre de classes des corps quadratiques imaginaires*, in Séminaire Bourbaki 1983–84, Astérisque **121–122**, Soc. Math. de France, 1985, pp. 309–323.

[**Oli1**] M. Olivier, *Corps sextiques primitifs*, Ann. Institut Fourier **40** (1990), 757–767.

[**Oli2**] M. Olivier, *The computation of sextic fields with a cubic subfield and no quadratic subfield*, Math. Comp. **58** (1992), 419–432.

[**Oli3**] M. Olivier, *Tables de corps sextiques contenant un sous-corps quadratique (I)*, Sém. Th. des Nombres Bordeaux (Série 2) **1** (1989), 205–250.

[**Oli4**] M. Olivier, *Corps sextiques contenant un corps quadratique (II)*, Sém. Th. des Nombres Bordeaux (Série 2) **2** (1990), 49–102.

[**Oli5**] M. Olivier, *Corps sextiques contenant un corps cubique (III)*, Sém. Th. des Nombres Bordeaux (Série 2) **3** (1991), 201–245.

[**Oli6**] M. Olivier, *Corps sextiques primitifs (IV)*, Sém. Th. des Nombres Bordeaux (Série 2) **3** (1991), 381–404.

[**Ore**] Ö. Ore, *Newtonsche Polygone in der Theorie der algebraischen Körper*, Math. Ann. **99** (1928), 84–117.

[**Poh1**] M. Pohst, *On the computation of number fields of small discriminants including the minimum discriminants of sixth degree fields*, J. Number Theory **14** (1982), 99–117.

[**Poh2**] M. Pohst, *A modification of the LLL-algorithm*, J. Symb. Comp. **4** (1987), 123–128.

[**Poh3**] M. Pohst, *On computing isomorphisms of equation orders*, Math. Comp. **48** (1987), 309–314.

[**Poh4**] M. Pohst, *A note on index divisors*, in [PPWZ], 1991, pp. 173–182.

[**Poh-Wei-Zas**] M. Pohst, P. Weiler and H. Zassenhaus, *On effective computation of fundamental units I and II*, Math. Comp. **38** (1982), 275–329.

[**Poh-Zas1**] M. Pohst and H. Zassenhaus, *Über die Berechnung von Klassenzahlen und Klassengruppen algebraische Zahlkörper*, J. Reine Angew. Math. **361** (1985), 50-72.

[**Pol1**] J. Pollard, *Theorems on factorization and primality testing*, Proc. Cambridge Phil. Soc. **76** (1974), 521–528.

[**Pol2**] J. Pollard, *A Monte-Carlo method for factorization*, BIT **15** (1975), 331–334.

[**Pom**] C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, in [MCC], 1983, pp. 89–139.

[**Quer**] J. Quer, *Corps quadratiques de 3-rang 6 et courbes elliptiques de rang 12*, C.R. Acad. Sci. Paris **305** (1987), 1215–1218.

[**Rab**] M. Rabin, *Probabilistic algorithms for testing primality*, J. Number Theory **12** (1980), 128–138.

[**Rib**] K. Ribet, *On modular representations of* $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ *arising from modular forms*, Invent. Math. **100** (1990), 431–476.

[**Rub**] K. Rubin, *Tate-Shafarevitch groups and L-functions of elliptic curves with complex multiplication*, Invent. Math. **93** (1987), 527–560.

[**von Schm1**] J. Graf v. Schmettow, *Beiträge zur Klassengruppenberechnung*, Dissertation, Univ. Düsseldorf, 1991.

[**von Schm2**] J. Graf v. Schmettow, *KANT – a tool for computations in algebraic number fields*, in [PPWZ], 1991, pp. 321–330.

[**Schn**] C.P. Schnorr, *A more efficient algorithm for lattice basis reduction*, J. Algorithms **9** (1988), 47–62.

[**Schn-Euch**] C.P. Schnorr and M. Euchner, *Lattice basis reduction: Improved practical algorithms and solving subset sum problems*, Proc. of the FCT 1991, LN in Comp. Sci. **529**, Springer-Verlag, Berlin, Heidelberg, 1991, pp. 68–85.

[**Schn-Len**] C.P. Schnorr and H.W Lenstra, *A Monte-Carlo factoring algorithm with linear storage*, Math. Comp. **43** (1984), 289–312.

[**Schön**] A. Schönhage, *Probabilistic computation of integer polynomial GCD*, J. Algorithms **9** (1988), 365–371.

[**Scho**] R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **43** (1985), 483–494.

[**Scho2**] R. Schoof, *Counting points of elliptic curves over finite fields*, J. Th. des Nombres Bordeaux (Série 2) **7** (1995), 219–254.

[**SPD**] A. Schwarz, M. Pohst and F. Diaz y Diaz, *A table of quintic number fields*, Math. Comp. **63** (1994), 361–376.

[**Sel-Wun**] J. Selfridge and M. Wunderlich, *An efficient algorithm for testing large numbers for primality*, Proc. Fourth Manitoba Conf. Numer. Math. (1974), 109–120.

[**Ser1**] J.-P. Serre, *Sur les représentations modulaires de degré 2 de* $\mathrm{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$, Duke Math. J. **54** (1987), 179–230.

[**Sey1**] M. Seysen, *A probabilistic factorization algorithm with quadratic forms of negative discriminants*, Math. Comp. **48** (1987), 757–780.

[**Sey2**] M. Seysen, *Simultaneous reduction of a lattice basis and its reciprocal basis*, Combinatorica **13** (1993), 363–376.

[**Sha1**] D. Shanks, *Class number, a theory of factorization, and genera*, Proc. Symp. in Pure Maths. **20**, A.M.S., Providence, R.I., 1969, pp. 415–440.

[**Sha2**] D. Shanks, *On Gauss and composition I and II*, Number theory and applications, R. Mollin (ed.), Kluwer Academic Publishers, 1989, pp. 163–204.

[**Sha3**] D. Shanks, *The infrastructure of a real quadratic field and its applications*, Proc. 1972 Number theory conference, Boulder (1972), 217–224.

[**Sha4**] D. Shanks, *Incredible identities*, Fibon. Quart. **12** (1974).

[**Sha-Wil**] D. Shanks and H. Williams, *A note on class number one in pure cubic fields*, Math. Comp. **33** (1979), 1317–1320.

[**Shi1**] G. Shimura, *On the zeta-function of an Abelian variety with complex multiplication*, Ann. of Math. **94** (1971), 504–533.

[**Shi2**] G. Shimura, *On elliptic curves with complex multiplication as factors of the Jacobians of modular function fields*, Nagoya Math. J. **43** (1971), 199–208.

[**Sie**] C.L. Siegel, *Über die Classenzahl quadratischer Zahlkörper*, Acta Arith. **1** (1935), 83–86.

[**Sil1**] R. Silverman, *The multiple polynomial quadratic sieve*, Math. Comp. **48** (1987), 329–340.

[**Sil2**] J. Silverman, *Computing heights on elliptic curves*, Math. Comp. **51** (1988), 339–358.

[**Soi**] L. Soicher, *The computation of Galois groups*, Thesis, Concordia Univ., Montreal, 1981.

[**Soi-McKay**] L. Soicher and J. McKay, *Computing Galois groups over the rationals*, J. Number Theory **20** (1985), 273–281.

[**Sol-Str**] R. Solovay and V. Strassen, *A fast Monte-Carlo test for primality*, SIAM J. Comput. **6** (1977), 84–85; erratum ibid. **7** (1978), p. 118.

[**Star**] H. Stark, *Class numbers of complex quadratic fields*, in Modular Functions of one variable I, LN in Math. **320**, Springer-Verlag, Berlin, Heidelberg, 1973, pp. 153–174.

[**Stau**] R.P. Stauduhar, *The determination of Galois groups*, Math. Comp. **27** (1973), 981–996.

[**Tay-Wil**] R. Taylor and A. Wiles, *Ring-theoretic properties of certain Hecke algebras*, Ann. of Math. **141** (1995), 553–572.

[**Ten-Wil**] M. Tennenhouse and H. Williams, *A note on class number one in certain real quadratic and pure cubic fields*, Math. Comp. **46** (1986), 333-336.

[**Tra**] B. Trager, *Algebraic factoring and rational function integration*, Proceedings of SYMSAC '76 (1976), 219–226.

[**Val**] B. Vallée, *Une approche géométrique des algorithmes de réduction en petite dimension*, Thesis, Univ. of Caen, 1986.

[**Wag**] C. Wagner, *Class number 5, 6 and 7*, Math. Comp. **65** (1996), 785–800.

[**de Weg**] B. de Weger, *Algorithms for Diophantine equations*, Dissertation, Centrum voor Wiskunde en Informatica, Amsterdam, 1988.

[**Weil**] A. Weil, *Number of solutions of equations in finite fields*, Bull. A.M.S. **55** (1949), 497–508.

[**Wie**] D. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Information Theory **32** (1986), 54–62.

[**Wiles**] A. Wiles, *Modular elliptic curves and Fermat's last theorem*, Ann. of Math. **141** (1995), 443–551.

[**Wil-Jud**] H. Williams and J. Judd, *Some algorithms for prime testing using generalized Lehmer functions*, Math. Comp. **30** (1976), 157–172 and 867–886.

[**Wil-Zar**] H. Williams and C. Zarnke, *Some algorithms for solving a cubic congruence modulo p*, Utilitas Math. **6** (1974), 285–306.

[**Zag1**] D. Zagier, *On the values at negative integers of the zeta-function of a real quadratic field*, Ens. Math. **22** (1976), 55–95.

[**Zag2**] D. Zagier, *Modular forms whose Fourier coefficients involve zeta-functions of quadratic fields*, Modular functions of one variable VI, LN in Math. **627**, Springer-Verlag, Berlin, Heidelberg, New-York, 1977, pp. 105–169.

[**Zag3**] D. Zagier, *Large integral points on elliptic curves*, Math. Comp. **48** (1987), 425–436.

[**Zim1**] R. Zimmert, *Ideale kleiner Norm in Idealklassen und eine Regulatorabschätzung*, Invent. Math. **62** (1981), 367–380.

[**Zip**] R. Zippel, *Simplification of expressions involving radicals*, J. Symb. Comp. **1** (1985), 189–210.

Index

Q

R

S