Advanced computational number theory

MHT 933

2009-2010

Foreward

The following is essentially a set of notes written last years by Pr. Karim Belabas who was responsible of teaching *Computational Number Theory* to "Master 2 Research" students.

I have only

- removed some comments and examples, essentially in the first chapter;
- added some sections here and there in the following ones;
- completed some missing lectures;
- developed some proofs;
- corrected some mistakes;
- written a specific set of exercises, available on my webpage.

Thanks a lot to him for his great work!

As said at the presentation done in september, the course uses classical and modern factorization algorithms to present important ideas and techniques in computational number theory. We will cover the reduction of $\mathbb{Z}$-modules and lattices, factorization of univariate polynomials over finite fields, the rationals and the complex numbers, then primality testing (up to the Elliptic Curve Primality Proving algorithm) and integer factorization (up to the Number Field Sieve).

The emphasis is on important ideas throughout, and asymptotically fast methods, certainly not programming efficiency. Many tricks must be implemented before most of the algorithms we will study become really practical, and blazingly fast as they are meant to be. For instance, many algorithms will compete to achieve a given result, each with a certain range of input sizes on which it will be optimal, in a given environment. Hence, a good implementation should provide all of them, as well as finely tuned thresholds to decide which method to use. We shall ignore such concerns and blissfully lose constant or logarithmic factors when technical details would obscure our main point.

Very good references covering about the same material are Gerhard & von zur Gathen [24] for chapters 1–3, Cohen [6] for chapter 2, 4, 5 and Crandall & Pomerance [9] for chapter 4.

This is work in progress, that may contain mistakes. Please send any suggestion of improvement to `Jean-Paul.Cerri@math.u-bordeaux1.fr`

Happy reading!

Jean-Paul Cerri

Contents

Chapter 1

Introduction

1.1 Basic definitions

1.1.1 Algorithms

Algorithms give an answer that is sure.
For computations that just give a highly probable result, we shall use the word *methods*.
Algorithms are classified in two categories, the *randomized* or *probabilistic* ones (using random generators) and the *deterministic ones*.

1.1.2 Running time

Running the program expends resources: time, space (memory), etc.
Unless mentionned otherwise, the resource we are interested in is *running time*.
In general we have to garantee that less than $f(s)$ units of resources will be spent, where all instances have input size $\leq s$. The quantity $f(s)$ can be the number of arithmetic operations (*arithmetic* or *algebraic* complexity) or the number of bit operations (*binary* or *word* complexity).
This running time or cost $f(s)$ can be defined essentially in two ways: in the *worst case* or *on average*. When not precised, it will ever be in the worst case.
In the case of a randomized algorithm we shall speak of *expected cost* for the cost on average: for a fixed input i, we average the cost over all possible runs of the program, i.e. if the set of possible runs is the finite set S_i and the cost

of a given run $a \in S_i$ is $f_i(a)$, we compute

$$E(f_i) = \frac{1}{\sharp S_i} \sum_{a \in S_i} f_i(a).$$

The expected cost for a size s is the max of the $E(f_i)$ over all i of size at most s. Of course input size s must be defined precisely and suitably for each particular problem. In general we shall use the O notation for $f(s)$.

- if $f(s) = O(s)$ the algorithm is said *linear time*;
- if $f(s) = s^{O(1)}$ it is said *polynomial time*;
- if $f(s) = O(\exp(s^c))$ it is said *subexponential* if $c < 1$ and *exponential* otherwise.

We shall often use the soft-O notation: $\widetilde{O}(f) = O(f) \times (\log f)^{O(1)}$.

1.1.3 Examples

Example 1.1. Let $n \in \mathbb{Z}_{>0}$ be an integer needing at most $s \geq 1$ binary digits to be encoded (or bits to be stored). We have

$$n = \sum_{i=0}^{s-1} a_i 2^i,$$

where $a_i \in \{0, 1\}$ for all i and $(a_0, \ldots, a_{s-1}) \neq (0, \ldots, 0)$. Let $m \in \mathbb{Z}_{>0}$ be another integer needing at most s digits to be encoded. The computation of $m + n$ has word complexity $O(s)$ and the naive computation of mn (with all products bit by bit) has word complexity $O(s^2)$ (*quadratic* time algorithm).

Example 1.2. Let $n \in \mathbb{Z}_{>0}$ and R be a ring. If P and Q are two elements of $R[X]$ and have degree less than n, the computation of $P + Q$ has arithmetic complexity $O(n)$ and the naive computation of PQ (with all products coefficient by coefficient) has arithmetic complexity $O(n^2)$.

Example 1.3. A randomized algorithm (which gives a correct answer with our convention) is also called *Las Vegas* method. A probabilistic method which only gives a probable result is called *Monte Carlo* method. For instance n being given, computing $a^{n-1} \mod n$ for $k \geq 1$ integers a chosen uniformly at random with $1 < a < n$ can show that n is composite if we do not obtain 1 for some a. But if we have $a^{n-1} \equiv 1 \mod n$ for all the a tested, we cannot say anything else than "n is maybe prime". In fact, if n is a Carmichael number (composite), it will be the case if all the a tested are coprime to n, which is probable with probabilty $(\phi(n)/n)^k = \prod_{p|n}(1 - 1/p)^k$.

1.2 Some principles and examples

1.2.1 Arithmetic is hard, Linear Algebra is easy

A prototypical example: given a, $b \in \mathbb{Z}_{>0}$, compute $\gcd(a, b)$. Factoring a and b is a hard problem. Nevertheless, if $a = bq + r$ where q, $r \in \mathbb{Z}$, then $\gcd(a, b)$=$\gcd(b, r)$, so that we can use Euclidean division to garantee $0 \leq r < b$ and iterate. This can be written as

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -q \end{pmatrix} = \begin{pmatrix} b & r \end{pmatrix},$$

which gives

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -q_1 \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_k \end{pmatrix} = \begin{pmatrix} \gcd(a, b) & 0 \end{pmatrix}.$$

We just use Euclidean divisions and conceptually a sequence of matrix multiplications, which can give us a Bezout relation! Fast and easy!

1.2.2 Be Lazy

Avoid all work that is not absolutely necessary and defer any costly computation until it is impossible to avoid it.

- optimizing the inner loops;
- precomputing as much as possible;
- work with formal symbols (formal computation);
- sparse representations: in this case, computations can be simplified;
- approximate computations and reconstruction;

Example 1.4. (square and multiply) To compute $x^{17} \bmod p$ see that

$$x^{17} \bmod p = ((((x^2)^2)^2)^2).x \bmod p$$

which needs 5 modular multiplications instead of 16 (with the same complexity).

Example 1.5. (Horner scheme) Let R a polynomial of $A[X]$ (A commutative ring). We have

$$R(t) = R_0 + t(R_1 + t(\cdots(R_{n-2} + R_{n-1}t)\cdots))$$

which gives an evaluation in $2n$ better than in $3n$ as in the naive approach (computation of the t^i, etc).

Let us now develop the last point.

Example 1.6. Let $A \in M_n(\mathbb{Z})$, we want to compute $\det A$. The usual method by Gauss pivoting is very expensive in the sense that it introduces rational coefficients whose size increases after each matrix operation. Possible solutions are:

1. compute with floating point numbers and round final result. Problem of stability.

2. take a bound $M(A)$ for $|\det A|$, for instance the trivial one

   $$M(A) = n! \max_{i,j} |a_{i,j}|^n$$

   or better the Hadamard's bound

   $$M(A) = \sqrt{\prod_{i=1}^{n} \Big(\sum_{j=1}^{n} a_{i,j}^2 \Big)}.$$

   Then choose a prime $p > 2M(A)$ and compute $\det \overline{A}$ in $M_n(\mathbb{F}_p)$ which is equal to $\det A$ mod p. Since $\det A \in [-M(A), M(A)] \subseteq (-p/2, p/2)$ there is a single value possible for $\det A$. Problem: find a great prime.

3. take distinct primes $p_1, \ldots, p_k$ whose product is $> 2M(A)$. Compute the determinant of A modulo p_i for each i and use the Chinese Remainder Theorem to solve the k congruences that we obtain. This is the most efficient approach.

This trick that consists in such a modular approach is called an *homomorphic imaging scheme*: map to cheaper rings, compute there, then come back.

1.2.3 Divide and conquer: Karatsuba and generalization.

Let us begin with an elementary example. Imagine that we want to compute the product of two polynomials P, $Q \in R[X]$ of degree $< n$, where R is a commutative ring. We have already seen that the naive approach leads to an arithmetic complexity in $O(n^2)$. A way of improving this result is to proceed in the following way. First of all we consider our polynomials as “polynomials” of “degree” $< 2^s$ where s is the smallest integer such that $n \leq 2^s$, i.e. $s = \lceil \log n / \log 2 \rceil$. Suppose $s > 0$. We write

$$P = X^{2^{s-1}} P_1 + P_2 \quad \text{and} \quad Q = X^{2^{s-1}} Q_1 + Q_2,$$

where P_1, P_2, Q_1 and Q_2 are polynomials of degree $< 2^{s-1}$. Then we have

$$\begin{aligned} PQ &= X^{2^s}P_1Q_1 + X^{2^{s-1}}(P_1Q_2 + P_2Q_1) + P_2Q_2 \\ &= X^{2^s}P_1Q_1 + X^{2^{s-1}}\Big\{(P_1+P_2)(Q_1+Q_2) - P_1Q_1 - P_2Q_2\Big\} + P_2Q_2, \end{aligned}$$

so that we have just to compute 3 products $A = P_1Q_1$, $B = P_2Q_2$, $C = (P_1 + P_2)(Q_1+Q_2)$ of polynomials of degree $< 2^{s-1}$. The sum $Y = X^{2^s}P_1Q_1+P_2Q_2$ being just a concatenation, the computation of P_1+P_2, Q_1+Q_2 needing each at most 2^{s-1} additions and the computation of $A+B$, $Z = C-(A+B)$ and $Y + X^{2^{s-1}}Z$ needing each at most 2^s additions (or substractions), we have:

$$C(s) \leq 3C(s-1) + 2^{s+2},$$

where $C(s)$ is the arithmetic complexity for the computation of the product of two polynomials of degree $< 2^s$ $(s \geq 0)$. As $C(0) = 1$ an elementary computation leads to $C(s) \leq 9.3^s - 2^{s+3}$. Recalling that $s \sim \log n/\log 2$, this allows to use a recursive algorithm with arithmetic complexity in $O(n^{\log 3/\log 2})$ which is better than $O(n^2)$.

Remark 1.7. Instead of cutting in two parts we can cut in three, four,…, k parts. This is the Toom-Cook's method which gives in theory a complexity in $O(n^{\log(2k-1)/\log k})$ but is difficult to use for great values of k ! See Exercise 2.

Here is a pratical lemma which allows to compute easily complexity when using a divide and conquer approach.

Lemma 1.8. *Let* $f : \mathbb{R}^+ \to \mathbb{R}^+$ *such that*

- f *is bounded on* $[0, 1)$*;*
- $f(x) \leq af(x/b) + Mx$ *for some* a, $M > 0$ *and* $b > 1$.

then

$$f(x) = \begin{cases} O(x^{\log a/\log b}) & \textit{if} \quad a > b \\ O(x\log x) & \textit{if} \quad a = b \\ O(x) & \textit{if} \quad a < b \end{cases}$$

Proof. Suppose $x \geq 1$ and let k be the smallest integer such that $x/b^k < 1$, i.e. $k = \lfloor \log x/\log b \rfloor + 1$. Then we have by a trivial induction

$$f(x) \leq a^k f\Big(\frac{x}{b^k}\Big) + M\Big(\frac{a^{k-1}}{b^{k-1}} + \cdots + \frac{a}{b} + 1\Big)x.$$

This gives

$$f(x) \leq la^k + Mx\Big(\frac{a^{k-1}}{b^{k-1}} + \cdots + \frac{a}{b} + 1\Big),$$

where l is an upper bound for f on $[0,1)$. But

$$a^k \in \Big[a^{\frac{\log x}{\log b}}, a^{\frac{\log x}{\log b}+1}\Big] = \Big[x^{\frac{\log a}{\log b}}, ax^{\frac{\log a}{\log b}}\Big]$$

and

$$\frac{a^{k-1}}{b^{k-1}} + \cdots + \frac{a}{b} + 1 = \begin{cases} k & \text{if} \quad a = b \\ \frac{(a/b)^k - 1}{a/b - 1} & \text{if} \quad a \neq b \end{cases}$$

In the case where $a < b$, this last quantity is bounded by $b/(b-a)$ and we get finally an $O(x)$.
In the case where $a = b$, we find that

$$f(x) \leq l \max(a,1)x + kMx = O(x \log x).$$

In the case where $a > b$, we have

$$\frac{(a/b)^k - 1}{a/b - 1} \leq b\frac{(a/b)^{(\log x/\log b)+1}}{a-b} \leq a\frac{x^{(\log a/\log b)-1}}{a-b},$$

which finally gives an $O(x^{\log a/\log b})$. □

This result can be generalized.

Lemma 1.9. *Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ such that*

- *f is bounded on $[0,1)$;*
- *$f(x) \leq af(x/b) + Mx^r$ for some a, M, $r > 0$ and $b > 1$.*

then

$$f(x) = \begin{cases} O(x^{\log a/\log b}) & \textit{if} \quad a > b^r \\ O(x^r \log x) & \textit{if} \quad a = b^r \\ O(x^r) & \textit{if} \quad a < b^r \end{cases}$$

Proof. Put $g(x) = f(x^{1/r})$ and use Lemma 1.8. □

Example 1.10. Coming back to Karatsuba, we see that we can define f by $f(x) = C(s)$ if $x \in [2^{s-1}, 2^s)$ for $s \in \mathbb{Z}_{>0}$ and $f(x) = C(0) = 1$ if $x \in [0,1)$. It is easy to see that we have conditions 1 and 2 of Lemma 1.8 with $a = 3$, $b = 2$ and $M = 8$, because $C(s) \leq 3C(s-1) + 2^{s+2}$ if $s > 0$ and $1 \leq 3 + 8x$ if $x \in [0,1)$. Lemma 1.8 gives $f(x) = O(x^{\log 3/\log 2})$ which is coherent with our previous evaluation because the complexity for degree $< n$ is $\leq f(n)$ by definition of f.

1.2.4 Fast Fourier Transform

We now give an important application of the previous principle. Let us consider polynomials of $A[X]$ where A is a commutative ring with unity 1. For simplicity suppose that 2 is invertible in A. Let $n = 2^k$ with $k > 0$ so that n is invertible too. Let $\omega \in A$ be an n-th primitive root of 1 ($\omega^n = 1$ and $\omega^d - 1$ is not a zero divisor if $1 \leq d < n$). We admit that A admits such a root. We call *Discrete Fourier Transform* of a polynomial $R \in A[X]$ of degree $< n$, identified with the n-tuple $(R_0, \dots, R_{n-1})$ the n-tuple

$$DFT_\omega(R) = (R(1), R(\omega), \dots, R(\omega^{n-1})) \in A^n.$$

Let us put $m = n/2 = 2^{k-1}$. For $0 \leq p < m$ we have

$$R(\omega^p) = \sum_{j=0}^{m-1} R_{2j}\alpha^{jp} + \omega^p \sum_{j=0}^{m-1} R_{2j+1}\alpha^{jp}$$

and

$$R(\omega^{p+m}) = \sum_{j=0}^{m-1} R_{2j}\alpha^{jp} - \omega^p \sum_{j=0}^{m-1} R_{2j+1}\alpha^{jp},$$

where $\alpha = \omega^2$ is an m-th primitive root of 1. This follows essentially from $\omega^m = -1$ (because $(\omega^m + 1)(\omega^m - 1) = 0$ and $\omega^m - 1$ is not a zero divisor). This leads to the following recursive algorithm to compute $DFT_\omega(R)$.

Algorithm 1. Fast DFT

Input: R, ω, the $(\omega^j)_{j<n/2}$ are precomputed.
Output: $DFT_\omega(R)$.

```
$m = n/2$
$S = (R_0, R_2, \dots, R_{n-2})$ and $T = (R_1, R_3, \dots, R_{n-1})$
$u = DFT_{\omega^2}(S)$
$v = DFT_{\omega^2}(T)$
for $p$ from 0 to $m - 1$ do
    $z_p = \omega^p v_p$; $w_p = u_p + z_p$; $w_{p+m} = u_p - z_p$.
Return $w$
```

Let us analyze the complexity of this algorithm. It is easy to see that

$$c_k = 2c_{k-1} + 3.2^{k-1} \quad \text{and} \quad c_0 = 0,$$

where c_k is the complexity for degree $< 2^k$. Then putting $d_k = c_k - (3/2)k2^k$, we have $d_k = 2d_{k-1}$ and since $d_0 = 0$ we have $d_k = 0$ and finally

$$c_k = \frac{3}{2}k2^k.$$

Finally we obtain an arithmetic complexity in $(3/2)k2^k = 3/(2\log 2)n\log n = O(n\log n)$. Note that we could also have used Lemma1.8 with $f(x) \leq 2f(x/2) + 3x$ in the same way as for Karatsuba.

Compare to the obvious approach where we evaluate successively the $R(\omega^i)$. We compute $R(t)$ in linear time thanks to Horner scheme, yelding a quadratic algorithm for the DFT.

This leads to an improvement of the algorithms already seen for the computation of the product of two polynomials, thanks to the following lemma.

Lemma 1.11. *For every $R \in A[X]$ of degree $< n$ we have*

$$DFT_{\omega^{-1}}(DFT_\omega(R)) = DFT_\omega(DFT_{\omega^{-1}}(R)) = nR.$$

Proof. Let $Q = DFT_\omega(R) = (R(1), R(\omega), \dots, R(\omega^{n-1}))$. We have $DFT_{\omega^{-1}}(Q) = (Q(1), Q(\omega^{-1}), \dots, Q(\omega^{-n+1}))$ and its k-th coordinate $(0 \leq k < n)$ is

$$\sum_{j<n}\Big(\sum_{i<n} R_i\omega^{ij}\Big)\omega^{-kj} = \sum_{i<n} R_i\Big(\sum_{j<n}\omega^{j(i-k)}\Big).$$

From

$$\Big(\sum_{j<n}\omega^{(i-k)j}\Big)(1-\omega^{i-k}) = 1-\omega^{(i-k)n} = 0,$$

and since $1-\omega^d$ is not a zero divisor if $0 < d < n$, we have

$$\sum_{j<n}\omega^{(i-k)j} = 0$$

if $i \neq k$. Finally the k-th coordinate $(0 \leq k < n)$ of $DFT_{\omega^{-1}}(Q)$ is nR_k. □

This gives a new algorithm to compute PQ where P, $Q \in A[X]$ are of degrees $< n/2$.

Algorithm 2. Fast multiplication in $A[X]$

```
Input: $P$, $Q \in A[X]$ of degrees $< n/2$.
Output: $PQ$.
Compute $DFT_\omega(P) = (a_0, a_1, \dots, a_{n-1})$
Compute $DFT_\omega(Q) = (b_0, b_1, \dots, b_{n-1})$
Compute $(c_0, c_1, \dots, c_{n-1}) = (a_0b_0, a_1b_1, \dots, a_{n-1}b_{n-1})$
Return $1/nDFT_{\omega^{-1}}(C)$ where $C = \sum c_iX^i$.
```

Proof. We have

$$(c_0, c_1, \dots, c_{n-1}) = (P(1)Q(1), P(\omega)Q(\omega), \dots, P(\omega^{n-1})Q(\omega^{n-1})) = DFT_\omega(PQ)$$

where PQ is of degree $< n$. From Lemma 1.11

$$\frac{1}{n}DFT_{\omega^{-1}}(C) = \frac{1}{n}DFT_{\omega^{-1}}(DFT_\omega(PQ)) = PQ.$$

□

Corollary 1.12. *Provided A contains a primitive root of 1 of degree $2n = 2^{k+1}$ and 2 is invertible in A, polynomials of $A[X]$ of degree $< n$ can be multiplied in $O(n \log n)$ operations in A (in fact $(9/\log 2)n \log n + O(n)$).*

Proof. With $m = 2n$, our three DFT need $(9/2 \log 2)m \log m$ ring operations and the computation of C needs m products. This gives $(9/\log 2)n \log 2n + 2n = (9/\log 2)n \log n + O(n)$ ring operations. □

Corollary 1.13. *Suppose that A supports FFT (contains 2^k-th primitive roots of 1 for every k) and that 2 is invertible in A. Then, for every n, polynomials of $A[X]$ of degree $< n$ can be multiplied in $O(n \log n)$ operations in A (in fact $(18/\log 2)n \log n + O(n)$ because we have to take $2^k \geq n$).*

Finally we have still improved (asymptotically) the complexity of Karatsuba's or Toom-Cook's algorithms.

1.2.5 Schönage-Strassen

But what to do if A does not admit a 2^k-th primitive root of 1 ? To simplify again, we can first suppose that 2 is invertible in A. We shall see later what to do when it is not the case. Let $n = 2^k$ and imagine that we want to compute PQ where $P, Q \in A[X]$ are of degree $< n/2$. For that we can compute the product $P''Q''$ of two polynomials of degree $< t$ whose coefficients are in a ring with $2t$-th primitive root.
Let $m = 2^{\lfloor k/2 \rfloor}$ and $t = n/m = 2^{\lceil k/2 \rceil}$. Write P and Q as

$$P = \sum_{i<t} P_i X^{mi} \quad \text{and} \quad Q = \sum_{i<t} Q_i X^{mi},$$

where P_i, $Q_i \in A[X]$ have degree $< m$. Let us put

$$P' = \sum_{i<t} P_i Y^i \quad \text{and} \quad Q' = \sum_{i<t} Q_i Y^i.$$

P' and Q' are elements of $A[X, Y]$ and

$$PQ(X) = P'Q'(X, X^m).$$

So we are done if we can compute $P'Q'$. We put $B = A[X]/(X^{2m}+1)$ and we note $\omega = (X \bmod X^{2m}+1) \in B$. We put $P'' = P'$ modulo $X^{2m}+1$ and $Q'' = Q'$ modulo $X^{2m}+1$, i.e. $P'' = \sum_i P_i(\omega)Y^i$ and $Q'' = \sum_i Q_i(\omega)Y^i$, which are in $B[Y]$ with degree $< t$ and we compute $P''Q''$ in $B[Y]$ by FFT. Computing this product can be done via FFT because there is a $2t$-th primitive root of 1 in B: $\omega' = \omega$ if $t = 2m$ or $\omega' = \omega^2$ if $t = m$ (indeed ω is a $4m$-th primitive root). Substituing X^m instead of Y and X instead of ω gives the result. See why.

Example 1.14. Let $A = \mathbb{F}_5$, $P = X^4 + 2X + 3$, $Q = 2X^3 + X^2 + 4X + 2$. We have $n = 8$, $m = 2$, $t = 4$, there is no 8-th primitive root of 1 in A and 2 is invertible in A. Here we have $P' = Y^2 + 2X + 3$, $Q' = Y(2X+1) + 4X + 2$ and if $\omega = X \bmod X^4+1$ ($\in B = A[X]/(X^4+1)$) we have $P'' = Y^2 + 2\omega + 3$, $Q'' = Y(2\omega+1) + 4\omega + 2$. Since ω is a 8-th primitive root in B, we can compute $P''Q''$ in $B[Y]$ via FFT. This gives :

$$P''Q'' = Y^3(2\omega+1) + Y^2(4\omega+2) + Y(4\omega^2+3\omega+3) + 3\omega^2 + \omega + 1,$$

which yelds

$$\begin{aligned} PQ &= X^6(2X+1) + X^4(4X+2) + X^2(4X^2+3X+3) + 3X^2 + X + 1 \\ &= 2X^7 + X^6 + 4X^5 + X^4 + 3X^3 + X^2 + X + 1. \end{aligned}$$

Now, what to do if 2 is not invertible in A ? A small changing in the algorithm allows to compute in fact $2^k PQ$. Another modification of the previous algorithm (with 3-adic DFT instead of 2-adic) gives $3^s PQ$ for some s. From a Bezout relation $u2^k + v3^s = 1$ in $\mathbb{Z}$ we recover the product PQ.

The final result is

Theorem 1.15. *(Cantor-Kaltofen). Over any commutative ring A, polynomials of degree $< n$ can be multiplied in time $O(n \log n \log\log n) = \widetilde{O}(n)$ operations in A.*

Corollary 1.16. *Two positive integers less than 2^n represented by bit-strings can be multiplied in time $O(n \log n \log\log n) = \widetilde{O}(n)$.*

Proof. In order to multiply $a = \sum_{i<n} a_i 2^i$ and $b = \sum_{i<n} b_i 2^i$ (digits in $\{0,1\}$), multiply the polynomials $\sum a_i X^i$ and $\sum b_i X^i$ in the stated time bound. Then evaluate the result at 2, starting from the lower degree terms. The coefficients of the product polynomial are $\leq n+1$, so the evaluation handles $O(\log n)$ bits each time a new coefficient is considered, for a negligible $O(n \log n)$ total cost. □

1.3 Elementary complexity results

1.3.1 In $\mathbb{Z}$

The size of an integer a is the number of bits required to store a, i.e $s(a) = \lfloor \log_2(a) \rfloor + 1$. Assume all operands have size less than n.

Operations	Naive	Fast
$a+b$	$O(n)$	$O(n)$
$a \times b$	$O(n^2)$	$\widetilde{O}(n)$
$a = bq + r$	$O(n^2)$	$O(M_{\mathbb{Z}}(n))$
Extended gcd	$O(n^2)$	$O(M_{\mathbb{Z}}(n) \log n)$
CRT	$O(n^2)$	$O(M_{\mathbb{Z}}(n) \log n)$

- $M_{\mathbb{Z}}(n)$ is the multiplication time in $\mathbb{Z}$ for two operands of size less than n. The fast algorithm is based on Schönhage-Strassen multiplication, in time $O(n \log n \log \log n)$.

- for the Euclidean division, the input is (a, b), $b \neq 0$, and the output (q, r) with $0 \leq r < |b|$. The fast algorithm solves the equation $b - a/x = 0$ using the Newton Iteration

  $$x_{n+1} = x_n - x_n(x_n b - a).$$

  (Let the precision increase with the iterations and use fast multiplication.) We then set $q = \lfloor x \rfloor$, then $r = a - bq$. The complexity stated assumes than $M_{\mathbb{Z}}(n)$ satisfies properties like $M(n)/n \geq M(m)/m$ for all $n \geq m$, and $M(mn) \leq m^2 M(n)$, it is in particular applicable for the Schönhage-Strassen and the naive quadratic multiplication.

- in the Extended gcd, the input consists of two integers a, b and the output consists of the $\gcd(a, b)$ and two integers u, v such that $au + bv = \gcd(a, b)$. The fast gcd is based on the divide an conquer paradigm and is quite technical.

- CRT stands for Chinese remainder algorithm where the input consists of n congruences $x \equiv a_k \pmod{b_k}$ where the b_k are pairwise coprime with and the output expected is a solution for the above congruences. We assume $s(a_k) \leq s(b_k)$ and $\sum_k s(b_k) \leq n$. The fast algorithm uses three divide-and-conquer passes: first to compute a product tree, then all modular inverses simultaneously, then a standard recursion.

1.3.2 In $\mathbb{Z}/N\mathbb{Z}$

We choose a canonical representative in each congruence class. A natural choice are the integers in $[0, N-1]$; another is $]-N/2, N/2]$, which is often more efficient when we need small negative integers, but a little more complicated to describe. In both cases, the size of any input is less than $s(N)$.

An addition is implemented as an addition in $\mathbb{Z}$, possibly followed by a subtraction. Multiplication, is a multiplication in $\mathbb{Z}$, followed by a division by N. Inversion is an extended gcd followed by a multiplication. So the costs are the same as in $\mathbb{Z}$, except for fast division which is more expensive by a factor $\log n$.

1.3.3 In $K[X]$ where K is a field

Here the costs for operations in $K[X]$ counts the number of operations in K (we may multiply by the cost of an elementary operation in K when the latter is fixed). The operations taken into account are $+$, $-$, $\times$, $/$ in K. Let f, g be two polynomials in $K[X]$. If $h \in K[X]$, the size of h is $S(h) = mdegh{+}1 \leq n$.

Operation	Naive	Fast
$f+g$	$O(n)$ $[+]$	$O(n)$ $[+]$
$f \times g$	$O(n^2)$ $[+, \times]$	$\widetilde{O}(n)$
$f = gh + r$	$O(n^2)$	$O(M_{K[X]}(n))$
Extended gcd	$O(n^2)$	$O(M_{K[X]}(n) \log n)$
CRT	...	$O(M_{K[X]}(n) \log n)$

1.3.4 In $K[X]/(T)$

Important special cases are finite field extensions $\mathbb{F}_q/\mathbb{F}_p$, and finite extensions of $\mathbb{Q}$. As in $\mathbb{Z}/N\mathbb{Z}$ we work with polynomials of size $\leq s(T)$, so the costs are as above. Again, fast modular division is slower by a factor $\log n$ than fast Euclidean division.

1.3.5 In $M_{n \times n}(K)$:

Again, we count the operations in K, for $A \in M_{n\times n}(K)$, $S(A) = n^2$.

Operations	Naive	Fast
$A+B$	$O(n^2)$	$O(n^2)$
$A \times B$	$O(n^3)$	$O(n^\omega), \omega = 2.376$
$A = LU$	$O(n^3)$	$O(n^\omega)$

The *LU* factorization is enough to solve most linear algebra problems over K: computing kernels, image, rank profile... In the above, ω is called a *feasible multiplication exponent*. The best value used for practical sizes is $\omega = \log_2 7 \approx 2.8$ (Strassen).

Black Box Linear Algebra: in this model, costs are calculated as the number of evaluations $x \mapsto Ax$ for a "black box matrix" A. (The name comes from the fact that we do not know anything about A except how it acts on vectors: it is an opaque operator, or a black box.) In general, matrix-vector multiplication is an $O(n^2)$ operation but most matrices encountered in practice have some structure which make evaluation cheaper, e.g. diagonal or band matrices, sparse matrix (as in the factorbase algorithms used to factor integers), Sylverster's matrix from the resultant, Berlekamp matrix (used to factor polynomials over finite fields), FFT matrix (= van der Monde on roots of unity), etc.

In this model, on can compute the *LU* factorisation of A in $O(n)$ evaluations and $O(n^2)$ field operations. So we gain nothing on general matrices, but quite a lot for special matrices. Contrary to the $O(n^{2.376})$ method, this is quite practical.

Chapter 2

Lattices

2.1 $\mathbb{Z}$-modules

2.1.1 Definitions

- Any abelian group G, its law of composition written additively, can be made into a module over $\mathbb{Z}$ in exactly one way, by the rules $0 \cdot g = 0_G$, $n \cdot g = g + \cdots + g$ (n times) and $(-n) \cdot g = -(n \cdot g)$ for any integer $n > 0$. We may thus identify abelian groups and $\mathbb{Z}$-modules, as well as submodules with subgroups.

- More generally, for any $n > 0$, we may define an action of $\mathbb{Z}^n$ on G^n by right multiplication:

$$(g_1, \ldots, g_n) \cdot \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = \sum_{i=1}^{n} \lambda_i g_i.$$

- If $A \subset G$, the submodule/subgroup generated by A is

$$\langle A \rangle_{\mathbb{Z}} := \Big\{ \sum_{i \in I} \lambda_i a_i,\ (\lambda_i) \in \mathbb{Z}^I,\ (a_i) \in A^I,\ I \text{ finite} \Big\}$$

- G is of *finite type* if $G = \langle A \rangle_{\mathbb{Z}}$ with A finite. In this case, we have $G = A \cdot \mathbb{Z}^{\#A}$; in other words, any element of G is of the form $A \cdot \lambda$. All our modules will be of this type.

- A family $g = (g_1, \ldots, g_n)$ is *free* (linearly independent) if and only if

$$g \cdot \lambda = 0,\ \lambda \in \mathbb{Z}^n \Rightarrow \lambda = 0.$$

- G (of finite type) is *free* if and only if it has a basis $(g_1, \dots g_n)$ which is free and generates G, i.e. $\langle g_1, \dots, g_n \rangle_{\mathbb{Z}} = G$

Example 2.1. Not all modules have a basis. $\mathbb{Z}/2\mathbb{Z}$ is not free because 2 times anything is 0, and $2 \neq 0$, so no non-empty subset of $\mathbb{Z}/2\mathbb{Z}$ can be free. However, $\mathbb{Z}^n$ is free.

We state without proof two basic theorems about modules (actually valid over principal rings, not only $\mathbb{Z}$).

Theorem 2.2 (Adapted Basis)**.** *Let G be a free $\mathbb{Z}$-module of finite type, H a submodule. There exists a basis $(g_1, \dots, g_n)$ of G and $d_n \mid \cdots \mid d_1$, $d_i \in \mathbb{Z}_{\geq 0}$ (note that d_i can be zero), such that $\{d_i g_i,\ d_i > 0\}$ is a basis for H. In particular, H is free.*

The integers $d_1, \dots, d_n$ are well-defined: they do not depend on the basis (g_i).

Corollary 2.3 (Elementary Divisors)**.** *Let G be a $\mathbb{Z}$-module of finite type. There exists $g_1, \dots, g_n$ in G such that*

$$\begin{aligned} G &= \bigoplus_{i=1}^{n} (\mathbb{Z}/d_i\mathbb{Z}) \cdot g_i, \quad \textit{where} \quad d_n \mid \cdots \mid d_1, \quad d_i \in \mathbb{Z}_{\geq 0}. \\ &= \underbrace{\bigoplus_{i=1}^{r} \mathbb{Z} \cdot g_i}_{\mathbb{Z}^r} \oplus \underbrace{\bigoplus_{r+1}^{n} (\mathbb{Z}/d_i\mathbb{Z}) \cdot g_i}_{G_{tor}} \end{aligned}$$

where r is defined as the rank of G, G_{tor} is the torsion group of G and contains all the elements of finite order.

The meaning of the direct sum is:

$$\sum_{i=1}^{n} \lambda_i g_i = 0, \quad \lambda_i \in \mathbb{Z} \Leftrightarrow \lambda_i = d_i\mathbb{Z}, \quad \forall i.$$

We will make this explicit using linear algebra over $\mathbb{Z}$.

2.1.2 Hermite Normal Form (HNF)

Studying free modules is similar to studying vector spaces. If L is a free submodule of rank n in some $\mathbb{Z}^m$, it can be represented as an $m \times n$ matrix whose columns give the coordinates of a basis of L on the canonical basis of $\mathbb{Z}^m$. The representation is not unique, because it depends on a choice of basis for L. In vector spaces, L can be brought to column echelon form using

Gaussian elimination, but we cannot divide over $\mathbb{Z}$. Instead, analogous forms in $\mathbb{Z}$-modules are the Hermite normal form and the Smith normal form.

The Hermite Normal form generalizes the Gauss-Jordan form (over fields) to modules. The algorithm was a home exercise from the first lectures. Here is a comparison with 2×2 matrices. If $a \neq 0$, Gaussian elimination yields:

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} 1 & -b/a \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} a & 0 \end{pmatrix}$$

Using the Euclidean algorithm instead, we obtain:

$$\begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} u & s \\ v & t \end{pmatrix} = \begin{pmatrix} \delta & 0 \end{pmatrix}$$

where $\delta = \gcd(a,b)$, $s = -b/\delta$, $t = a/\delta$ and u and v satisfy the Bezout relation $au + bv = \delta$. The multiplying matrix is in $\mathrm{SL}_2(\mathbb{Z})$.

Definition 2.4. The matrix $(0 \mid H)$ is in *Hermite Normal Form* (HNF) if $H = (H_{ij})$ is an $m \times r$ matrix of maximal rank $r \leq n$ such that there exists a strictly increasing function

$$f : \{1, \dots, r\} \to \{1, \dots, m\}$$

satisfying

1. $q_j := H_{f(j),j} > 0$ and $H_{i,j} = 0$ if $i > f(j)$,
2. $0 \leq H_{f(j),k} < q_j$ if $k > j$.

It is easier to tell what the definition is saying with a picture of the matrix:

$$\left(\begin{array}{ccc|ccccc}
0 & \cdots & 0 & q_1 & \times & \times & \times & \times \\
0 & \cdots & 0 & & \times & \times & \times & \times \\
 & & & & \times & \times & \times & \times \\
 & & & & q_2 & \times & \times & \times \\
 & & & & & q_3 & \times & \times \\
 & & & & & & q_4 & \times \\
 & & & & & & & \times \\
0 & \cdots & 0 & & & & & q_5
\end{array}\right)$$

The matrix has m rows and n columns, $n - r$ of which are zero and r of which are nonzero (here $r = 5$). The × entries can be zero, positive or negative. There are two conditions:

- $f(j)$ is the row where the pivot $q_j > 0$ lies. The function f is called the rank profile because it tells the rank and where the pivot may be found: the matrix $\left(H_{f(i),j}\right)_{i,j\leq r}$ is non-singular.
- All the coefficients to the right of a pivot q_j are reduced mod q_j. So the $\times$ to the right of a q_j are non-negative.

Example 2.5. Assume $H \in M_{n\times n}(\mathbb{Z})$ has rank n. In this (simplest) case $r = n$ and the rank profile f is the identity.

Theorem 2.6. *The set of $m \times n$ HNF matrices form a system of representatives of $M_{m\times n}(\mathbb{Z}) / \mathrm{GL}_n(\mathbb{Z})$.*

N.B. $\mathrm{GL}_n(\mathbb{Z})$ acts on $M_{m\times n}(\mathbb{Z})$ by right multiplication. The previous formulation is equivalent to the following one: if $A \in M_{m\times n}(\mathbb{Z})$, there exist a unique $(0 \mid H) \in M_{m\times n}(\mathbb{Z})$ in HNF and a matrix $U \in GL_n(\mathbb{Z})$ (not necessary unique) such that $(0 \mid H) = AU$.

Corollary 2.7. *If we fix a basis of a free module $G \simeq \mathbb{Z}^m$, any submodule of G has a canonical basis: the one given by an HNF matrix. We shall speak of an HNF-basis.*

Proof. It is sufficient to see that the submodule G' can be represented by a matrix whose n columns are elements of one of its bases (described by their coordinates in the fixed basis of G). Since these bases are defined modulo $GL_n(\mathbb{Z})$, the unicity of the HNF gives the result. □

Remark 2.8. Note that if A is the matrix associated not only to a basis of G' but more generally to a generating family of G', the HNF of A will be the same if we do not take care of the zero-columns.

2.1.3 Smith Normal Form (SNF)

Definition 2.9. A matrix $(0 \mid D)$ or $\begin{pmatrix} 0 \\ D \end{pmatrix}$ is in *Smith Normal Form* (SNF) if D is diagonal, with diagonal $d_1, \ldots, d_n$ such that $d_n | \ldots | d_1$ in $\mathbb{Z}_{\geq 0}$.

Theorem 2.10 (Restatement of elementary divisors theorem)**.** *The set of $m \times n$ SNF matrices form a system of representatives of*

$$GL_m(\mathbb{Z}) \backslash M_{m\times n}(\mathbb{Z}) / GL_n(\mathbb{Z})$$

(Left and right multiplication respectively).

The previous formulation is equivalent to the following one: if $A \in M_{m\times n}(\mathbb{Z})$, there exist a unique $S \in M_{m\times n}(\mathbb{Z})$ in SNF and matrices $U \in GL_n(\mathbb{Z})$, $V \in GL_m(\mathbb{Z})$ (not necessary uniques) such that $S = VAU$.

2.1.4 Algorithms and Complexity

A good reference for these algorithms is Arne Storjohann's PhD dissertation (2000). See also Cohen which lacks details but is suitable for a quick implementation. Here are the input and output of the algorithms:

Input: $A \in M_{m\times n}(\mathbb{Z})$, size $nm\log(\max|a_{i,j}|)$.

Output (HNF): $H \in M_{m\times r}(\mathbb{Z})$, $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $AU = (0 \mid H)$ in HNF.

Output (SNF): $D \in M_{r\times r}(\mathbb{Z})$, $U \in \mathrm{GL}_n(\mathbb{Z})$, $V \in \mathrm{GL}_m(\mathbb{Z})$ such that $VAU = (0 \mid D)$ or $\begin{pmatrix} 0 \\ D \end{pmatrix}$ in SNF.

Below is a simple, but inefficient algorithm for HNF. There exist efficient, more complex, algorithms (see later). Note: to simplify book-keeping, we do not produce U (and swap columns so that indices are simpler to handle).

Algorithm 3. Naive Algorithm for HNF

Input: $A = (A_{i,j}) \in M_{m\times n}(\mathbb{Z})$

Output: $H \in M_{m\times r}(\mathbb{Z})$ such that $AU = (0 \mid H)$ in HNF.

```
Set $R \leftarrow 0$
for $i = m, m-1, \ldots, 1$ do    {line $i$}
  for $j = R+2, \ldots, n$ do    {zero $A_{i,j}$ using $A_{i,R+1}$}
    Write $\begin{pmatrix} A_{i,R+1} & A_{i,j} \end{pmatrix} \begin{pmatrix} u & s \\ v & t \end{pmatrix} = \begin{pmatrix} \delta & 0 \end{pmatrix}$    {Euclidean algorithm}
    $\begin{pmatrix} A_{*,R+1} & A_{*,j} \end{pmatrix} \leftarrow \begin{pmatrix} A_{*,R+1} & A_{*,j} \end{pmatrix} \begin{pmatrix} u & s \\ v & t \end{pmatrix}$    {$A_{*,j}$ : $j^{th}$ column}
  if $A_{i,R+1} \neq 0$ then
    $R \leftarrow R+1$
Reset $R \leftarrow 1$    {will increase up to 1+rank of matrix}
for $i = m, \ldots, 1$ do    {line $i$}
  if $A_{i,R} \neq 0$ then    {pivot; if no pivot, do nothing}
    Let $A_{i,*} \leftarrow A_{i,*} \times \mathrm{sign}(A_{i,R})$    {now $A_{i,R} > 0$}
    for $j = 1, \ldots, R-1$ do
      Let $q \leftarrow \lfloor A_{j,R}/A_{i,R} \rfloor$    {$A_{j,R} - qA_{i,R}$ is "reduced"}
      $A_{*,j} \leftarrow A_{*,j} - qA_{*,R}$
    let $R \leftarrow R+1$
Swap columns    {to get the un-mirrored HNF}
```

Remark 2.11. $AU = (0 \mid H)$, where U is uniquely determined if A is invertible. To recover U, apply the algorithm to the matrix $\begin{pmatrix} A \\ \mathrm{Id} \end{pmatrix}$ instead, but not for all its rows (this matrix is already in HNF), just for the m rows of A. See why.

The algorithm is made of two main **for** loops, one on lines 2 to 8, another on lines 10 to 19. The first loop uses the extended Euclidean algorithm to bring the matrix into left-upper triangular form. The last entries of row i will be $(A_{i,R+1}\ 0\ \ldots\ 0)$. The second loop reduces the entries to bring the matrix into mirrored HNF. The last step swaps the columns to obtain the true HNF.

Example 2.12. Let

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 2 & 4 \end{pmatrix}.$$

The first loop transforms successively A in $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 4 \end{pmatrix}$, then in $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ and finally in $\begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$, if we make successively the choices $\begin{pmatrix} 1 & 4 \\ 0 & -1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}$ and $\begin{pmatrix} 1 & 7 \\ 0 & -1 \end{pmatrix}$ for $\begin{pmatrix} u & s \\ v & t \end{pmatrix}$.

The second loop gives $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$, which leads (after swaping the columns) to the desired HNF

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

If now we apply the algorithm to the first two rows of

$$B = \begin{pmatrix} 2 & 3 & 1 \\ 1 & 2 & 4 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

we obtain

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 10 & 2 & -3 \\ -7 & -1 & 2 \\ 1 & 0 & 0 \end{pmatrix},$$

from which we deduce

$$\begin{pmatrix} 10 & 2 & -3 \\ -7 & -1 & 2 \\ 1 & 0 & 0 \end{pmatrix}$$

as a possible value for U. The value obtained for U, contrary to what happens with the HNF (which is unique), depends on the matrices $\begin{pmatrix} u & s \\ v & t \end{pmatrix}$ chosen in the algorithm. For instance, if instead of $\begin{pmatrix} 1 & 7 \\ 0 & -1 \end{pmatrix}$ in the last step of the first loop we take $\begin{pmatrix} 8 & 7 \\ -1 & -1 \end{pmatrix}$ (because we want $(1\ \ 7)\begin{pmatrix} u & s \\ v & t \end{pmatrix} = (1\ \ 0)$) we obtain

$$\begin{pmatrix} 10 & 12 & -3 \\ -7 & -8 & 2 \\ 1 & 1 & 0 \end{pmatrix}$$

for U, which is still a valid matrix.

A problem with the algorithm is that the size of entries increases during iterations. Measuring only algebraic complexity, the first loop requires $O(mnr)$ operations $(+, \times, \text{extended gcd})$ in $\mathbb{Z}$. The second loop requires $O(r^2 m)$ operations $(+, \times, \div)$ in $\mathbb{Z}$. Therefore, the overall order is $O(rm(n+r))$. If in addition you would like to recover U, replace the m by $(m+n)$ in the previous expression. The algorithm is *not* polynomial time, (size of input)$^{O(1)}$, because coefficient size increase too fast.

Kannan and Bachem (1979) found an algorithm that works in polynomial time. There are two main ideas.

- Reduce to the case where A is a square non-singular matrix of rank r. To find the rank profile, work over $\mathbb{Z}/p\mathbb{Z}$, for some suitable prime p.
- Work modulo $N := |\det A| \neq 0$. This prevents the previous problem of the entries blowing up, because at every step, all the entries belong to a fixed system of representatives of $\mathbb{Z}/N\mathbb{Z}$. The reason why it works is that the HNF of A and

$$\left(\begin{array}{c|ccc} & N & & 0 \\ A & & \ddots & \\ & 0 & & N \end{array}\right)$$

are identical.

You can reconstruct the HNF from this modular computation, see the proof in Cohen; it is a good exercise. Cohen does not explain how to get U; for this, see Storjohann.

Theorem 2.13. *Suppose $A \in M_n(\mathbb{Z})$ is nonsingular and let $B = \max|a_{i,j}|$.*

- *HNF can be solved in time* $\widetilde{O}\left(n^3 \mathrm{M}_{\mathbb{Z}}(n \log B)\right) = \widetilde{O}(n^4 \log B)$ *and space* $\widetilde{O}(n^3 \log B)$.

- *(Storjohann) HNF is solved in time* $\widetilde{O}(n^4 \log^3 B)$ *using space* $\widetilde{O}(n^2 \log B)$.

The first algorithm is fast, but requires a lot of memory. In the second, memory use is softly linear in the input size, essentially best possible.

Theorem 2.14. *SNF can be solved in polynomial time.*

Proof. Use the HNF algorithm on the rows then the columns of A, iterate. See why it gives effectively the SNF. □

Actually, SNF *without* U *and* V can be solved faster than HNF, provided we allow Monte-Carlo methods.

2.1.5 Applications

All these are applications of HNF and SNF.

- *Image* of a matrix $A \in M_{m \times n}(\mathbb{Z})$. We want to determine $A\mathbb{Z}^n$ which is a submodule of $\mathbb{Z}^m$ (the submodule generated by the columns of A) denoted by $\mathrm{Im}_{\mathbb{Z}}(A)$ to avoid ambiguity.

  Proposition 2.15. *If* $(0 \mid H)$ *is the* HNF *of* A*,* $\mathrm{Im}_{\mathbb{Z}}(A)$ *is the submodule of* $\mathbb{Z}^m$ *generated by the* r *independent columns of* H*. In particular the rank of* A *is* r*.*

  Proof. Obvious. □

  In example 2.12,

  $$\mathrm{Im}_{\mathbb{Z}}(A) = \mathbb{Z}\begin{pmatrix}1\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\1\end{pmatrix} = \mathbb{Z}^2.$$

- *Kernel* of A. We want to determine $\{x \in \mathbb{Z}^n \text{ such that } Ax = 0\}$ which is a submodule of $\mathbb{Z}^n$ denoted $\mathrm{Ker}_{\mathbb{Z}}(A)$ for similar reasons.

  Proposition 2.16. *If* $AU = (0 \mid H)$ *is the* HNF *of* A*, the* $n - r$ *first columns of* U *give a* $\mathbb{Z}$*-basis of* $\mathrm{Ker}_{\mathbb{Z}}(A)$*.*

Proof. Reading directly $(0 \mid H) = AU$ we see that, if U_i is the i-th column of U, we have $AU_i = 0$ for $1 \leq i \leq n - r$. Conversely let us consider a vector $X \in \mathrm{Ker}_{\mathbb{Z}}(A)$, i.e. $\in \mathbb{Z}^n$ such that $AX = 0$ and put $Y = U^{-1}X$. We have $AUY = 0$ with $Y \in \mathbb{Z}^n$, which yields $(0 \mid H)Y = 0$. Finally Y_i can be arbitrary for $i \leq n-r$ but the presence of the r pivots $H_{f(j),j} \neq 0$ for $1 \leq j \leq r$ leads to $Y_i = 0$ for $i > n - r$ and to the result when writing $X = UY$. □

In example 2.12,

$$\mathrm{Ker}_{\mathbb{Z}}(A) = \mathbb{Z}\begin{pmatrix} 10 \\ -7 \\ 1 \end{pmatrix}.$$

- *Equality* of two submodules of $\mathbb{Z}^m$. Let G_1 and G_2 be two submodules of $\mathbb{Z}^m$ defined thanks to $\mathbb{Z}$-bases g_1 and g_2 of same cardinality n. Then they are equal if and only if the HNF associated to the matrices A_1 and A_2 are equal, where A_i is the matrix in $M_{m\times n}(\mathbb{Z})$ whose j-th column is $(g_i)_j$. More generally if we have for each G_i a family $((g_i)_1, \ldots, (g_i)_{n_i})$ which generates G_i as a $\mathbb{Z}$-submodule of $\mathbb{Z}^m$, the G_i are equal if and only if the H-part of the HNF associated to A_1 and A_2 are equal, where $A_i \in M_{m\times n_i}$ is defined as above. See Remark 2.8.

- *Sum* of two submodules. Let G_1 and G_2 be two submodules of $\mathbb{Z}^m$ defined thanks to $\mathbb{Z}$-bases g_1 and g_2 of cardinalities n_1 and n_2. Let A_1 and A_2 be the two matrices defined as above. Then the HNF of $(A_1 \mid A_2)$ gives an HNF-basis for $G_1 + G_2$. Same thing with generating families instead of bases.

- *Inclusion* relation. Use $G_1 \subseteq G_2 \iff G_1 + G_2 = G_2$ and the two previous points.

- *Finite Abelian Groups.* Let G be such a group (or finite $\mathbb{Z}$-module). We know that
$$G \simeq \bigoplus_{1\leq i\leq n} (\mathbb{Z}/d_i\mathbb{Z})$$
with $d_n \mid \cdots \mid d_1$. Assume that we have bounds for the order of G, for instance that we know
$$a \leq \sharp G \leq b$$
with $b/a < 2$.

  By theoretical means we can find some integer m and a free $\mathbb{Z}$-module L of rank m such that G is isomorphic to L/L' where L' is a submodule of L of rank m but unknown.

We then determine as many elements of L' as possible so as to have at least m elements which are $\mathbb{Q}$-linearly independents.

We then compute the HNF-basis of L_1 which is the submodule of rank m generated by the elements that we have found.

Computing the determinant of this basis (which is trivial since the basis is in triangular form) already gives $\sharp L/L_1$. We can check whether $L_1 = L'$ (it is sufficient to have $a \leq \sharp L/L_1 \leq b$).

If not, we continue to find new elements of L' until the cardinality check shows thet $L_1 = L'$.

We can then compute the SNF of the HNF-basis and this gives us the complete structure of G.

This will be used later for the computation of the class groups.

2.2 Lattices

2.2.1 Definitions and first results

Let $E = (\mathbb{R}^n, q)$ be an Euclidean space, where q is a positive definite quadratic form. Let $\Lambda \subset \mathbb{R}^n$.

(Λ, q) is a lattice if Λ is a free $\mathbb{Z}$-module of rank n; the definition requires maximal rank, but this is only due to historical reasons, and to simplify notations later.
More generally: if Λ is just a free $\mathbb{Z}$-module (not embedded anywhere) of rank m, we can consider Λ embedded in $\mathbb{R}^m$:

$$\Lambda \subseteq \Lambda \otimes_{\mathbb{Z}} \mathbb{R} \simeq \mathbb{R}^m \quad \text{(dimension = rank)}$$

embedded via the function $x \mapsto x \otimes 1$. If $\Lambda = \langle b_1, \ldots, b_m \rangle_{\mathbb{Z}}$ with $b_i \in \mathbb{R}^n$, then $\Lambda = \{\sum_{i=1}^n \lambda_i b_i, \lambda_i \in \mathbb{Z}\}$, and we may identify

$$\Lambda \otimes \mathbb{R} = \{\sum_{i=1}^m \lambda_i b_i \colon (\lambda_i) \in \mathbb{R}^m\} = \langle b_1, \ldots, b_m \rangle_{\mathbb{R}}.$$

Then, even if we have a free $\mathbb{Z}$-module of smaller rank than dim E, we can consider it as a lattice in a smaller space.

Definition 2.17. A lattice (Λ, q) is a free $\mathbb{Z}$-module of finite rank n, together with a positive definite quadratic form on $\Lambda \otimes_{\mathbb{Z}} \mathbb{R} \simeq \mathbb{R}^n$.

Let $x \cdot y = \frac{1}{4}\big(q(x+y) - q(x-y)\big)$ be the scalar product associated to the quadratic form q. To any basis $(b_1, \ldots, b_n)$ we associate its Gram-Schmidt orthogonal basis $(b_1^*, \ldots, b_n^*)$, defined by

$$\begin{aligned} b_1^* &:= b_1, \\ b_i^* &:= b_i - \sum_{j<i} \mu_{i,j} b_j^*, \qquad 1 < i \leq n, \quad \text{where} \quad \mu_{i,j} := \frac{b_i \cdot b_j^*}{b_j^* \cdot b_j^*}. \end{aligned}$$

The recurrence formula follows from requiring that $b_i^* \cdot b_j^* = 0$ for $j < i$.

Note that if $\Lambda = \langle b_1, \ldots, b_n \rangle_{\mathbb{Z}}$ is a lattice, the $(b_i^*)_{i \leq n}$ do not lie in Λ in general since a priori the coefficients $\mu_{i,j}$ are not integers. The $(b_i^*)_{i \leq n}$ do however form an orthogonal $\mathbb{R}$-basis for $\mathbb{R}^n$, a priori not orthonormal. More generally, $\langle b_1^*, \ldots, b_r^* \rangle_{\mathbb{R}} = \langle b_1, \ldots, b_r \rangle_{\mathbb{R}}$ for any $1 \leq r \leq n$. This can be deduced by noting that the base change matrix is non-singular:

$$(b_1, \ldots, b_r) = (b_1^*, \ldots, b_r^*) \begin{pmatrix} 1 & \ldots & \mu_{1,r} \\ & \ddots & \vdots \\ 0 & & 1 \end{pmatrix} \tag{2.1}$$

Remark 2.18. From the Gram-Schmidt process, we could assume that q is the standard Euclidean form. Namely, we can set

$$\delta_i = \frac{b_i^*}{\sqrt{b_i^* \cdot b_i^*}}$$

to get an *orthonormal* basis. But for arithmetic applications, it is more flexible to retain the possibility of a general positive form. For instance, if $b_i \cdot b_j \in \mathbb{Z}$ for all i, j, then $\mu_{i,j} \in \mathbb{Q}$ for all i, j (proof by induction).

Let $E = (\mathbb{R}^n, q)$ be an Euclidean space, where $x \cdot x$ is the scalar product, and let Λ be a lattice with basis $(b_1, \ldots, b_n)$.

Definition 2.19.

- Let $\mathrm{Gram}(b_1, \ldots, b_n) := (b_i \cdot b_j)_{1 \leq i,j \leq n}$ the Gram matrix of the b_i.
- The discriminant of Λ is

$$\mathrm{disc}(\Lambda) := \det(\mathrm{Gram}(b_i)).$$

- The determinant of Λ is

$$\mathrm{d}(\Lambda) := \sqrt{\mathrm{disc}(\Lambda)}.$$

Proposition 2.20. *The discriminant* $\operatorname{disc}(\Lambda)$ *is well-defined and is equal to* $\prod_{i=1}^n q(b_i^*)$. *In particular, the latter depends only on the lattice and not on the chosen basis.*

Proof. Consider $(b_1^*, \ldots, b_n^*)$ the orthogonal basis of $\mathbb{R}^n$, such that $(b_i^*)A = (b_i)$, with $A \in \mathrm{Gl}_n(\mathbb{R})$ an upper triangular matrix with determinant 1 as in (2.1). Then

$$\mathrm{Gram}(b_1, \ldots, b_n) = A^T \mathrm{Gram}(b_1^*, \ldots, b_n^*) A.$$

Since $\mathrm{Gram}(b_1^*, \ldots, b_n^*)$ is diagonal, taking the determinant we obtain $\operatorname{disc}(\Lambda) = q(b_1^*) \ldots q(b_n^*)$. Now any other basis of Λ is of the form $(b_i') = (b_i)U$ for some $U \in \mathrm{Gl}_n(\mathbb{Z})$, replacing A by AU in the above. Since $\det U = \pm 1$, it follows that $\operatorname{disc}(\Lambda)$ is well-defined. □

Corollary 2.21 (Hadamard's inequality). *Let* $B \in M_n(\mathbb{R})$ *the matrix whose columns are some* $b_i \in \mathbb{R}^n$, *and let* $(\mathbb{R}^n, \| \cdot \|)$ *the standard Euclidean space. Then*

$$|\det B| = \prod_{i=1}^n \| b_i^* \| \leq \prod_{i=1}^n \| b_i \| .$$

Proof. Let Λ be the lattice generated by the b_i equiped with $q = \| \cdot \|^2$. We have

$$\operatorname{disc}(\Lambda) = \det(\mathrm{Gram}(b_i)) = \det(B^T B) = \det(B)^2.$$

But by the previous result we have

$$\operatorname{disc}(\Lambda) = \prod_{i=1}^n \| b_i^* \|^2,$$

so that

$$(\det B)^2 = \prod_{i=1}^n \| b_i^* \|^2 .$$

Moreover, since $b_i = b_i^* + \sum_{j<i} \mu_{i,j} b_j^*$ and since $(b_1^*, \ldots, b_i^*)$ is orthogonal we have

$$\| b_i \|^2 \geq \| b_i^* \|^2$$

and the final inequality. □

We are interested in short vectors. We shall now see that short vectors do exist, where "short" only depends on the dimension and the discriminant of the lattice. But this theorem does not say how to find them.

2.2.2 Minkowski's Theorem

Theorem 2.22 (Minkowski). *Let C be a subset of $\mathbb{R}^n$ such that:*

- *C is symmetric ($C = -C$),*
- *C is convex,*
- $\mathrm{vol}(C) > 2^n \mathrm{d}(\Lambda)$*;*

then there is a non-zero lattice point in C.

In the theorem, $\mathrm{vol}(C)$ is the volume with respect to the Euclidean volume form, i.e. the Lebesgue measure if q is the standard form. More generally, if A is the base change matrix expressing an orthonormal basis of E in terms of the canonical basis, the volume form is the Lebesgue measure divided by $|\det(A)|$.

Lemma 2.23. *If* $\mathrm{vol}(C) > \mathrm{d}(\Lambda)$*, then there exists c_1, $c_2 \in C$, $c_1 \neq c_2$ such that $c_1 \equiv c_2$* $\mathrm{mod}\ \Lambda$*.*

Proof. (of Lemma) Let $(b_i)_{1 \leq i \leq n}$ a basis of Λ and $\mathcal{F}$ be the fundamental domain for Λ (that is a complete system of representatives for $\mathbb{R}^n/\Lambda$) given by:

$$\mathcal{F} = \left\{ \sum_{i=1}^{n} \lambda_i b_i, 0 \leq \lambda_i < 1 \right\}.$$

We have $\mathrm{vol}(\mathcal{F}) = \mathrm{d}(\Lambda)$. Let us define

$$C_x := (C - x) \cap \mathcal{F}, \quad \text{where} \quad x \in \Lambda.$$

Since $C - x$ is C translated, and translations conserve volumes,

$$\mathrm{vol}(C_x) = \mathrm{vol}(C \cap (\mathcal{F} + x)).$$

By construction the $\mathcal{F}+x$ are disjoint and cover $\mathbb{R}^n$: $\bigcup_{x\in\Lambda}(\mathcal{F}+x) = \mathbb{R}^n$. Now argue by contradiction: assume that the C_x are disjoint (if not, there exists x_1, $x_2 \in \Lambda$, $x_1 \neq x_2$ such that $C_{x_1} \cap C_{x_2} \neq \emptyset$. This leads to the existence of c_1, $c_2 \in C$ with $c_1 - x_1 = c_2 - x_2$ so that $c_1 \neq c_2$ and $c_1 \equiv c_2 \pmod{\Lambda}$ which proves the Lemma). Since

$$\mathcal{F} \supset \bigcup_{x\in\Lambda} C_x,$$

$$\mathrm{d}(\Lambda) = \mathrm{vol}(\mathcal{F}) \geq \sum_{x\in\Lambda} \mathrm{vol}(C_x) = \sum_{x\in\Lambda} \mathrm{vol}(C \cap (\mathcal{F}+x)) = \mathrm{vol}(C \cap \mathbb{R}^n) = \mathrm{vol}(C)$$

(the $\geq$ is obtained by disjointness). A contradiction. □

Proof. (Minkowski's theorem)

$$\mathrm{vol}\left(\frac{C}{2}\right) = \frac{\mathrm{vol}(C)}{2^n} > \mathrm{d}(\Lambda)$$

$$\begin{aligned}(\text{Lemma}) \Rightarrow \frac{c_1}{2} &= \frac{c_2}{2} + \lambda, \quad c_i \in C, \lambda \in \Lambda \setminus \{0\}\\ \Rightarrow \lambda &= \frac{c_1}{2} - \frac{c_2}{2} = \frac{1}{2}(c_1 - c_2)\end{aligned}$$

C is symmetric and convex, so $\frac{1}{2}(c_1 - c_2) \in C \setminus \{0\}$ and we are done. □

The proof is by contradiction, obviously ineffective.

Corollary 2.24 (Minkowski bis)**.** *Let C be a subset of $\mathbb{R}^n$ such that:*

- *C is compact,*
- *C is symmetric,*
- *C is convex,*
- $\mathrm{vol}(C) \geq 2^n \mathrm{d}(\Lambda)$*;*

then there is a non-zero lattice point in C.

Proof. Use Theorem 2.22 with $C_k = (1 + 1/k)C$, where $k \in \mathbb{Z}_{>0}$. Then there exists $x_k \in \Lambda \setminus \{0\}$ such that $x_k \in C_k \subseteq 2C$. But $2C$ being compact, we can extract from (x_k) a subsequence which is convergent. On the one hand, it is easy to see that its limit x is in every C_k (because the C_k are closed, decreasing and $x_k \in C_k$) so that it is in C (because C is closed). On the other hand, the lattice Λ being discrete, this subsequence is ultimately stationnary and we have the result. □

Corollary 2.25. *Let (Λ, q) be a lattice of rank n. There exists $x \in \Lambda \setminus \{0\}$ such that*

$$q(x) \leq \gamma_n \mathrm{disc}(\Lambda)^{\frac{1}{n}},$$

where γ_n only depends on n.

Proof. We may assume that the Euclidean space (E, q) is $(\mathbb{R}^n, \| \cdot \|^2)$ (why?). Let $C = \{x \in \mathbb{R}, \| x \| \leq R\}$, which is compact, convex and symmetric. In order to apply Corollary 2.24, we want $\mathrm{vol}(C) = \delta_n R^n \geq 2^n \mathrm{d}(\Lambda)$ where δ_n is the volume of the unit ball in $\mathbb{R}^n$. With the choice $R = 2\delta_n^{-1/n} \mathrm{d}(\Lambda)^{1/n}$, there exists $x \in \Lambda \setminus \{0\}$ such that $q(x) = \| x \|^2 \leq R^2 = 4\delta_n^{-2/n} \mathrm{disc}(\Lambda)^{1/n}$. □

2.2.3 Short vectors and LLL algorithm.

Our goal here is not only to find short vectors but also to construct bases of lattices with short vectors. In fact finding the "smallest vectors" is a very hard task so that solving the first problem is already difficult !

However, admit that we know how to solve the first problem and that we try to solve the second one. A naive idea would be to search for "smallest vectors" in the following way : take a smallest vector b_1, then search for a smallest vector b_2 with $b_1,\ b_2$ independent and iterate, hoping that this would give a basis. It is not a totally stupid idea because it works in dimension $n \leq 4$. More precisely, taking n independent vectors of minimal norm(s) (with the previous algorithm) when $n \leq 3$ always works, and for $n = 4$ there exists at least one basis that can be obtained in this way. But unfortunately, as soon as the dimension is ≥ 5, it does not work. Let us give a simple example.

Example 2.26. Let in $\mathbb{Z}^4$ equiped with the standard Euclidean form, the lattice

$$L = \mathbb{Z}\begin{pmatrix}2\\0\\0\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\2\\0\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\0\\2\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}1\\1\\1\\1\end{pmatrix}.$$

Among the shortest vecors of L there are b_1, b_2, b_3 and

$$b_4' = 2b_4 - b_1 - b_2 - b_3 = \begin{pmatrix}0\\0\\0\\2\end{pmatrix},$$

but (b_1, b_2, b_3, b_4') is not a basis for L: in fact

$$\langle b_1, b_2, b_3, b_4' \rangle_{\mathbb{Z}} = (2\mathbb{Z})^4$$

and $b_4 \notin \langle b_1, b_2, b_3, b_4' \rangle$. However, if we take b_4 instead of b_4' it is of course ok! Now consider in $\mathbb{Z}^5$ still equiped with the Euclidean form, the lattice

$$L = \mathbb{Z}\begin{pmatrix}2\\0\\0\\0\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\2\\0\\0\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\0\\2\\0\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}0\\0\\0\\2\\0\end{pmatrix} \oplus \mathbb{Z}\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}.$$

In this lattice the shortest vectors are $\pm b_1$, $\pm b_2$, $\pm b_3$, $\pm b_4$ and $\pm b_5'$ where $b_5' = 2b_5 - b_1 - b_2 - b_3 - b_4$, which are the only vectors with norm 2. As above, it is easy to see that we cannot obtain a basis for L with 5 of these vectors.

Our goal is to find a basis of short vectors for Λ, i.e. with $q(b_i)$ small. There are different ways to solve this problem and the most natural, coming directly from the above considerations are the Minkowski reduction and HKZ reduction (for Hermite-Korkine-Zolotarev). Even if we shall not go into details, we give the definitions of these reductions.

Definition 2.27 (Minkowski reduced basis)**.** A $\mathbb{Z}$-basis of Λ $(b_1, \dots, b_n)$ is said *Minkowski reduced* if for every i, b_i is a shortest vector (with respect to q) such that $(b_1, \dots, b_i)$ can be completed in a basis for Λ.

Definition 2.28 (HKZ reduced basis)**.** With the same notation, $(b_1, \dots, b_n)$ is *HKZ reduced* if

1. b_1 is a shortest vector of Λ (with respect to q)
2. If we project the b_i, $i > 1$ on the orthogonal of $\langle b_1 \rangle$ (with respect to q) denoted by F, we find elements of a HKZ reduced basis of a lattice Λ' of rank $n-1$ (equiped with the restriction of q to F).

The problem of the algorithms which can give such reductions is that they are exponential time so that we can use them only in small dimensions. We will eventually achieve our problem with the LLL (or L^3) algorithm. The name LLL comes from its three inventors: Arjen Lenstra, Hendrik Lenstra Jr., and László Lovász, in a landmark 1982 paper [16], with the intended application of factoring polynomials in $\mathbb{Q}[X]$.

We want to generalize the Euclidean algorithm, which iterates reductions $(a \leftarrow a \mod b)$ and swaps $(a \leftrightarrow b)$. Swap is easy to generalize to higher dimensions. What would be "reduction" ?

Size-reduction

Definition 2.29. We let $\lfloor x \rceil :=$ the nearest integer to $x = \lfloor x + \frac{1}{2} \rfloor$, then

$$a \operatorname{cmod} b := a - \left\lfloor \frac{a}{b} \right\rceil b.$$

(The letter c is for *center.*) We have $-\frac{|b|}{2} \leq a \operatorname{cmod} b < \frac{|b|}{2}$.

We want to define the meaning of "reduce $b_i \mod \langle b_1 \dots b_{i-1} \rangle_{\mathbb{Z}}$". The idea is to reduce the size of b_i by removing a linear combination of the $(b_1, \dots, b_{i-1})$. Which one ?

- Over $\mathbb{R}$, the meaning of "reduce $b_i \mod \langle b_1 \dots b_{i-1}\rangle_{\mathbb{R}}$" is clear: project on the orthogonal complement of $\langle b_1, \dots, b_{i-1}\rangle_{\mathbb{R}}$, i.e.
$$b_i \leftarrow b_i - \sum_{j=1}^{i-1} \mu_{i,j} b_j^*$$
- Over $\mathbb{Z}$ we could try:
$$b_i \leftarrow b_i - \sum_{j=1}^{i-1} \lfloor \mu_{i,j} \rceil b_j^*,$$
but this does not make sense: the b_j^* are not integral combinations of the b_i. Instead, we use the following reduction process: for all $j = i-1, \dots, 1$,
$$\text{let } b_i \leftarrow b_i - \lfloor \mu_{i,j} \rceil b_j, \quad \text{where } \mu_{i,j} = \frac{b_i \cdot b_j^*}{b_j^* \cdot b_j^*},$$
i.e the $\mu_{i,j}$ change as b_i does (the b_j^* do not!). This is the same idea as above: the vector $b = b_i - \mu_{i,j} b_j$ satisfy $b \cdot b_j^* = 0$, and doing this for all j by *decreasing* values we would end up in the orthogonal subspace to $\langle b_1 \dots b_{i-1}\rangle_{\mathbb{R}}$, in fact with the orthogonal projection of b_i there. Instead, because of our rounding $\mu_{i,j}$, we obtain some approximation. Note that the above is a sequence of elementary operations over $\mathbb{Z}$, hence invertible.

Lemma 2.30. *Let b_i the resulting vector. It satisfies the following:*
$$|\mu_{i,j}| \leq \frac{1}{2}, \quad \textit{for all } j < i.$$

Proof. Decreasing induction. Let $|\mu_{i,j}| \leq \frac{1}{2}$ for all j such that $\ell < j < i$. The transformation $b_i \leftarrow b_i - \lfloor \mu_{i\ell} \rceil b_\ell$ has a nice behaviour: it does not affect the $\mu_{i,j}$ for $j > \ell$ and replaces $\mu_{i,\ell}$ by $\mu_{i,\ell}$ cmod 1. The result follows. □

A vector b_i satisfying the condition in the Lemma is called *size-reduced*, with respect to the family $(b_1, \dots, b_{i-1})$.

Corollary 2.31. *Given any $\mathbb{Z}$-basis of Λ we may change it into a basis such that all the $|\mu_{i,j}| \leq \frac{1}{2}$ for every $j < i$.*

Remark 2.32. Since the lattice does not change, $\prod_{i=1}^n q(b_i^*)$ remains fixed. Further

- $q(b_1^*) = q(b_1)$, and more generally $q(b_i^*) \leq q(b_i)$.
- $q(b_i) = q(b_i^*) + \sum_{j<i} \mu_{i,j}^2 q(b_j^*)$, since the b_j^* are an orthogonal family.
- $\mu_{i,j}^2 \leq \frac{1}{4}$ so the b_i are short provided the $q(b_j^*)$ are.

Problem

If all $q(b_j^*)$ are small, so is $q(b_i)$ by the above and we are happy. But what occurs if $q(b_j^*)$ is much smaller than 1 for some j ? Since the product $\prod_{i=1}^n q(b_i^*) = \mathrm{disc}(\Lambda)$ is constant, then some other Gram-Schmidt vector b_j^* is very big, so at least one vector in the basis is very big: $q(b_j) \gg 1$, for some j.

So we want to avoid "tiny" Gram-Schmidt vectors. Ideally, we want the $q(b_i^*)$ to be roughly equal, of the order of $\mathrm{disc}(\Lambda)^{\frac{1}{n}}$ since their product is $\mathrm{disc}(\Lambda)$. We will swap vectors between reductions in order to ensure this.

We are now ready to give definitions.

Definition 2.33. The basis $(b_1, \dots, b_n)$ is *size reduced* if $|\mu_{i,j}| \leq \frac{1}{2}$ for every $j < i$.

To size-reduce a basis requires $O(n^2)$ operations. Recall we also want to avoid tiny $q(b_i^*)$:

Definition 2.34. $(b_i)_{i \leq n}$ is *Siegel reduced* if $q(b_i^*) \leq 2q(b_{i+1}^*)$ for $i < n$.

For such a basis, we have

$$q(b_1) = q(b_1^*) \leq 2q(b_2^*) \leq 2^2 q(b_3^*) \leq \cdots \leq 2^{n-1} q(b_n^*), \tag{2.2}$$

hence

$$q(b_1)^n \leq \prod q(b_i^*) \prod_{i<n} 2^i = \mathrm{disc}(\Lambda) \times 2^{n(n-1)/2}. \tag{2.3}$$

Proposition 2.35. *If $(b_i)_{i \leq n}$ is Siegel reduced, then*

$$q(b_1) \leq 2^{(n-1)/2} \mathrm{disc}(\Lambda)^{1/n}.$$

Compare with Minkowski and Corollary 2.25.

Definition 2.36. $(b_i)_{i \leq n}$ is called *reduced* if it is

- size reduced *and*
- Siegel reduced

In a Siegel reduced basis, the values $q(b_i^*)$ cannot decrease too fast, hence a small $q(b_i^*)$ means that *all* $q(b_j^*)$ are small for $j \leq i$. If the basis is further size-reduced, all b_i are small in that same range (Remark 2.32) and we are happy. From the Siegel condition, if a $q(b_i^*)$ is large (so that $q(b_i)$ is large), then all following ones are large also. Since the product is controlled, no really large vector can occur.

The LLL algorithm produces a reduced basis from an arbitrary basis. Before giving the algorithm, we describe the nice properties of a reduced basis:

Theorem 2.37. *Let $x \in \Lambda$, $x \neq 0$, and let $(b_i)_{i \leq n}$ be a reduced basis for Λ. Then the following three properties hold:*

1. $q(x) \geq \min_{i \leq n} q(b_i^*)$ *(this one actually holds for any basis)*

2. $q(b_1) \leq 2^{n-1} q(x)$ *(in other words, b_1 is essentially as short as possible)*

3. *(Generalization of the previous case.) Let $(x_1, \ldots, x_t)$ be t independent vectors in Λ; in particular, $x_i \neq 0$ and $t \leq n$. Then*
$$q(b_t) \leq 2^{n-1} \max_{j \leq t} q(x_j).$$

Remark 2.38. The second property gives in general a much better bound than that given by Minkowski's theorem or Proposition 2.35, since here the bound for $q(b_1)$ depends on the shortest x for a specific lattice, while Minkowski runs through all possible lattices with a given discriminant. Recall that according to Corollary 2.25, there is a non-zero vector x in Λ such that $q(x) \leq \gamma_n \mathrm{disc}(\Lambda)^{\frac{1}{n}}$, for some γ_n only depending on n.

Proof. We prove the 3 points in order.

1. Write $x = \sum_{i \leq n} \lambda_i b_i$, where $\lambda_i \in \mathbb{Z}$ and not all are 0 since $x \neq 0$. Let k be the maximal index such that $\lambda_k \neq 0$. Then
$$x = \sum_{i \leq k} \lambda_i (b_i^* + \sum_{j<i} \mu_{i,j} b_j^*) = \lambda_k b_k^* + \sum_{j<k} \nu_j b_j^* , \quad \nu_j \in \mathbb{R}.$$
Using the fact that $(b_1^*, \ldots, b_k^*)$ are orthogonal and $|\lambda_k| \geq 1$:
$$q(x) = \lambda_k^2 q(b_k^*) + \sum_{j<k} \nu_j^2 q(b_j^*) \geq q(b_k^*) \geq \min_{j \leq n} q(b_j^*).$$

2. $q(b_1) = q(b_1^*) \leq 2^{k-1} q(b_k^*) \leq 2^{k-1} q(x) \leq 2^{n-1} q(x)$.

3. (Thus far we have not made use of the assumption that the given basis is size reduced, only Siegel reduced. In practice however, decent algorithms giving a Siegel reduced basis yield a basis which is size reduced

too.) We start by generalizing (2.2), noting that Siegel reducedness implies that for $j < i$:

$$q(b_j^*) \leq 2q(b_{j+1}^*) \leq \cdots \leq 2^{i-j}q(b_i^*).$$

Next we prove the following inequality between vectors in the reduced basis and corresponding vectors in the orthogonal basis:

Lemma 2.39. *It holds that*

$$1 \leq \frac{q(b_i)}{q(b_i^*)} \leq 2^{i-1}$$

Proof. Start with the defining equation:

$$b_i = b_i^* + \sum_{j<i} \mu_{i,j} b_j^*.$$

Take the inner product of each side with itself and use the fact that the b_i^* are orthogonal:

$$q(b_i) = q(b_i^*) + \sum_{j<i} \mu_{i,j}^2 q(b_j^*).$$

Dividing through by $q(b_i^*)$ yields:

$$\frac{q(b_i)}{q(b_i^*)} = 1 + \sum_{j<i} \mu_{i,j}^2 \frac{q(b_j^*)}{q(b_i^*)}.$$

Note that the last term is non-negative. Now use $|\mu_{i,j}| \leq \frac{1}{2}$:

$$1 + \sum_{j<i} \mu_{i,j}^2 \frac{q(b_j^*)}{q(b_i^*)} \leq 1 + \sum_{j=1}^{i-1} \frac{1}{4} 2^{i-j} = 2^{i-2} + 2^{-1} \leq 2^{i-2} + 2^{i-2} = 2^{i-1}.$$

□

Now we prove our contention. Write the x_j in terms of the basis:

$$x_j = \sum_{i=1}^{n} r_{i,j} b_i\,, \quad r_{i,j} \in \mathbb{Z}.$$

For a fixed j, let $i(j)$ be the largest index i such that $r_{i,j} \neq 0$. Then by the proof of the first point (actually the second-last step) with $x = x_j$:

$$q(x_j) \geq q(b_{i(j)}^*)$$

By renumbering the x_j, we may assume that

$$i(1) \leq i(2) \leq \cdots \leq i(t)$$

We proceed to prove by induction that $j \leq i(j)$. Firstly $1 \leq i(1)$ since $x_1 \neq 0$. Now suppose $j-1 \leq i(j-1)$. Since $i(j-1) \leq i(j)$ we have $j-1 \leq i(j)$. But $j-1 = i(j)$ would imply that $\{x_1, \dots, x_j\}$ is contained in the subspace $\langle b_1, \dots, b_{j-1} \rangle$ spanned by the first $j-1$ basis vectors. This contradicts the assumption that the x_i are linearly independent. Hence $j-1 < i(j)$, or $j \leq i(j)$. Now combine the various little results:

$$q(b_j) \leq 2^{j-1} q(b_j^*) \leq 2^{j-1} 2^{i(j)-j} q(b_{i(j)}^*) = 2^{i(j)-1} q(b_{i(j)}^*) \leq 2^{n-1} q(x_j),$$

which completes the proof.

□

2.2.4 The LLL reduction algorithm

Let us give now the LLL-algorithm.

Algorithm 4. LLL algorithm

Input: (b_i) a $\mathbb{Z}$-basis for $\Lambda \subset \mathbb{R}^n$. We assume that the b_i are in $\mathbb{Z}^n$, and $\mathrm{Gram}(b_i) \in M_n(\mathbb{Z})$. (Hence the $\mu_{i,j}$ are in $\mathbb{Q}$.)

Output: A reduced basis for Λ.

```
let $k := 2$, compute the $(b_i^*)_{i \leq n}$   {i.e. compute $(\mu_{i,j})_{j<i\leq n}$.}
while $(k \leq n)$ do   {$(b_1, \dots, b_k)$ is reduced}
    Reduce $b_k$ mod $(b_1, \dots, b_{k-1})$, update $(b_i^*)_{i \leq n}$   {Reduction step}
    if $(k > 1)$ and $q(b_{k-1}^*) > 2q(b_k^*)$ then   {Swap if Siegel condition fails}
        Exchange $b_{k-1}$ and $b_k$
        Update $(b_i^*)_{i \leq n}$
        Set $k := k - 1$
    else
        Set $k := k + 1$
return $(b_1, \dots, b_n)$
```

Remark 2.40. It is not necessary to recompute the (b_i^*) from scratch each time a change is made: most do not change, and the others can be cheaply updated. See Cohen for formulas. There are many variants on Siegel reducedness, simpler to check and a little harder to explain, but yielding essentially the same bounds.

Remark 2.41. Since $(b_1, \ldots, b_n)$ is a basis, we can get the base change matrix going from the input basis to the output basis just from the input and output of the algorithm — this is easy linear algebra. What happens if the (b_i) are dependent vectors ? If the reduction of $b_k \bmod (b_1, \ldots, b_{k-1})$ is 0 we discard b_k, and everything else works. But in that case, from input/output, we lose some information: the kernel. If that is a problem, one can add more steps updating an auxiliary matrix (initially Id_n) to keep track of elementary operations, as usual.

Remark 2.42. We assume the b_i and $\mathrm{Gram}(b_i)$ are integral because we want exact arithmetic throughout. Using floating point arithmetic complicates quite a bit the analysis — we need perturbation results —, although it certainly improves efficiency in practice. This is not a serious theoretical restriction since we may approximate all entries by rationals then clear denominators.

Theorem 2.43. *Let $A = \max_{i \leq n} q(b_i)$. The algorithm stops after $O(n^4 \log A)$ operations on integers of size $O(n \log A)$. Thus it takes $\widetilde{O}(n^5 (\log A)^2)$ time and requires $O(n^3 \log A)$ space (since there are n^2 numbers of size $O(n \log A)$ each).*

Remark 2.44. In comparison, the advanced HNF algorithm takes $\widetilde{O}(n^4 (\log A)^3)$ time and $O(n^2 \log A)$ space. The LLL algorithm has the advantage however that it is easier to implement (coefficients do not blow up in a naive implantation), gives nicer base change matrices (smaller entries), and nicer output (bases whose vectors are provably small).

Proof. First step. Prove termination.

1. Some definitions: Let
$$\Lambda_i = \langle b_1, \ldots, b_i \rangle_{\mathbb{Z}}$$
denote the lattice in $\Lambda_i \otimes_{\mathbb{Z}} \mathbb{R}$ generated by the first i basis vectors. So $\Lambda_n = \Lambda$. Then define
$$D_i = \mathrm{disc}(\Lambda_i) = \prod_{j \leq i} q(b_j^*)$$
$$D := \prod_{i=1}^{n-1} D_i = q(b_1^*)^{n-1} \ldots q(b_{n-1}^*).$$
Then $D_i \in \mathbb{Z} \setminus \{0\}$, $i \leq n-1$, and hence $D \in \mathbb{Z}_{\geq 1}$. The idea is now to bound the maximum value that D can have at the beginning, and show that D only changes during a swap, becoming less than $\frac{3}{4} D$ every

time that it changes. This will bound the maximum number of swap steps, hence the number of iterations in the while loop. This proves termination, and will eventually give a bound on the running time of the algorithm, once we make sure coefficient explosion does not occur.

2. We now consider how D changes through the algorithm.

   Reduction of $b_k \mod (b_1, \ldots, b_{k-1})_{\mathbb{Z}}$ does not change the b_i^*, hence it leaves the D_i and hence D fixed.

   During the swap step (b_{k-1}^*, b_k^*) is replaced by (s, t), where

   $$s = b_k^* + \mu_{k,k-1} b_{k-1}^*$$

   since it is the component of b_k orthogonal to $b_1, \ldots, b_{k-2}$. In particular it follows that $q(s) \geq q(b_k^*)$, which will be useful later on. Since the product over all $q(b_i^*)$ stays constant, and only two of them change, we have

   $$q(s)q(t) = q(b_{k-1}^*)q(b_k^*),$$

   from which we deduce $q(t) \leq q(b_{k-1}^*)$. (Side note: t is the component of b_{k-1} orthogonal to $b_1, \ldots, b_{k-2}, b_k$, which spans a subspace containing $b_1, \ldots, b_{k-2}$. But b_{k-1}^* is defined as the component of b_{k-1} orthogonal to this latter subspace, hence $q(t) \leq q(b_{k-1}^*)$.)

   Summing up, the D_j are unaffected except D_{k-1}, which gets multiplied by $q(s)/q(b_{k-1}^*)$. From the expression for s we get:

   $$q(s) = q(b_k^*) + \mu_{k,k-1}^2 q(b_{k-1}^*)$$

   $$\implies \frac{q(s)}{q(b_{k-1}^*)} = \frac{q(b_k^*)}{q(b_{k-1}^*)} + \mu_{k,k-1}^2 \leq \frac{1}{2} + \frac{1}{4} = \frac{3}{4},$$

   using the fact that we are swapping precisely because the Siegel condition was not satisfied at (b_{k-1}^*, b_k^*), and the size condition.

   So this proves that during each swap D gets replaced by an integer which is less than $\frac{3}{4}D$.

3. From the definitions of A and D it follows that at the beginning of the algorithm

   $$D \leq q(b_1)^{n-1} \ldots q(b_{n-1}) \leq A^{\frac{n(n-1)}{2}}.$$

   By the above it follows that the number of swaps is bounded by $\log_{\frac{3}{4}} A^{\frac{n(n-1)}{2}}$, or more neatly, it is $O(n^2 \log A)$. Thus there are $O(n^2 \log A)$ loops, and since each loop involves $O(n^2)$ operations on the coordinates of the b_i and $\mu_{i,j}$, the total number of operations is $O(n^4 \log A)$, as claimed.

Second step. Bound the denominators.

Lemma 2.45. *At any point in the algorithm the following are true:*

1. $D_{k-1}b_k^* \in \mathbb{Z}^n$
2. $D_\ell \mu_{k,\ell} \in \mathbb{Z}$, *for all* $\ell < k \leq n$.

Proof. 1. We can express

$$b_k^* = b_k - \sum_{\ell<k} \lambda_{k,\ell} b_\ell$$

for certain $\lambda_{k,\ell} \in \mathbb{R}$. Recall that $b_k^* \cdot b_j = 0$ for $j < k$. Taking the inner product of every term in the above equation with b_j for j varying between 1 and $k-1$ gives the following $k-1$ linear equations in $k-1$ variables:

$$\sum_{\ell<k} \lambda_{k,\ell}(b_\ell \cdot b_j) = (b_k \cdot b_j) \ , \quad j < k.$$

The determinant of the linear system is D_{k-1}, hence using Cramer's rule to solve it, we deduce that $D_{k-1}\lambda_{k,\ell} \in \mathbb{Z}$, as claimed.

2. Using the definitions and what was just proved gives the following:

$$D_\ell \mu_{k,\ell} = D_{\ell-1} q(b_\ell^*) \frac{b_k \cdot b_\ell^*}{q(b_\ell^*)} = b_k \cdot (D_{\ell-1} b_\ell^*) \in \mathbb{Z}$$

□

Thus a bound for the D_i will give a bound for all the denominators. To uniformly bound the D_i, start by recalling that A was chosen such that $q(b_i) \leq A$ for all i at the beginning of the algorithm. This implies that $q(b_i^*) \leq A$ for all i at the beginning. Now we claim that $\max_i q(b_i^*)$ never increases, i.e. it is bounded throughout by A. This has essentially already been shown, since these values do not change except at the swap step, where both $q(s)$ and $q(t)$ were shown to be less than $q(b_{k-1}^*) \leq A$ by induction. Thus:

$$D_i = \prod_{j \leq i} q(b_i^*) \leq A^i.$$

Hence all the D_i are bounded by A^n. Then it follows from the previous step that the denominators of the b_k^* and $\mu_{k,l}$ are bounded by A^n, and since the b_i are anyway integers, this bounds all denominators of rational numbers in the algorithm by $O(n \log A)$.

Third step. Bound the absolute values of the entries. Using the previous step, this will bound the absolute values of the numerators.

1. We claim that
$$|\mu_{i,j}|^2 \leq D_{j-1}q(b_i).$$
This follows from the definitions and the Cauchy-Schwartz inequality:
$$\begin{aligned} |\mu_{i,j}|^2 &= \left(\frac{b_i \cdot b_j^*}{q(b_j^*)}\right)^2 \leq \frac{q(b_i)}{q(b_j^*)} \\ q(b_j^*) &= \frac{D_j}{D_{j-1}} \geq \frac{1}{D_{j-1}} \end{aligned}$$

2. Next we need to bound the $q(b_i)$. The idea is to show that $q(b_i) \leq nA$ everywhere except possibly during the reduction step where b_i is being reduced, where the weaker bound $q(b_i) \leq n^2(4A)^{n+1}$ still holds. At the beginning $q(b_i) \leq A \leq nA$ by definition. The only place in the algorithm where $q(b_i)$ changes is at the reduction step when b_i is being reduced. To bound $q(b_i)$ during this step and at the end, we use the bound $q(b_i^*) \leq A$ and the expression
$$b_i = b_i^* + \mu_{i,i-1}b_{i-1}^* + \cdots + \mu_{i,1}b_1^*.$$
Letting $\mu_{i,i} = 1$ and using orthogonality gives:
$$q(b_i) = \sum_{j \leq i} |\mu_{i,j}|^2 q(b_j) \leq \sum_{j \leq i} m_i^2 A \leq n m_i^2 A,$$
where
$$m_i = \max\{|\mu_{i,j}|;\ 1 \leq j \leq i\}$$
At the end of the reduction step $m_i = 1$ since $|\mu_{i,j}| \leq \frac{1}{2}$ for $j < i$ and 1 for $j = i$. Thus the bound $q(b_i) \leq nA$ always holds outside the reduction step.

   To bound $q(b_i)$ during the reduction step, we need a bound for the m_i which holds throughout the reduction step (we fix some i from now on). We give a bound for m_i which holds at the beginning of the step, and then show that m_i does not grow too much during the reduction step. So at the beginning of the reduction step:
$$\begin{aligned} m_i^2 &= \max_i\{|\mu_{i,j}|^2;\ 1 \leq j \leq i\} \\ &\leq \max_i\{D_{j-1}q(b_i);\ 1 \leq j \leq i\} \\ &\leq A^{n-1}q(b_i) \leq A^{n-1}nA = nA^n \end{aligned}$$

Now consider how m_i changes during one step in the loop of the reduction step for some $1 \leq j < i$, i.e. when b_i is being replaced by $b_i - \lfloor \mu_{i,j} \rceil b_j$. As mentioned in the beginning of the lecture, after this step the new values of $\mu_{i,l}$ for $j \leq l < i$ are less than $\frac{1}{2}$ in absolute value. Thus they will not have an incremental effect on the new value of m_i. For $1 \leq l < j$ the following holds for the new value of $\mu_{i,l}$ (which is given by the first expression):

$$\begin{aligned}\frac{(b_i - \lfloor \mu_{i,j} \rceil b_j) \cdot b_l^*}{b_l^* \cdot b_l^*} &= |\mu_{i,l} - \lfloor \mu_{i,j} \rceil \mu_{j,l}| \\ &\leq |\mu_{i,l}| + |\lfloor \mu_{i,j} \rceil| \cdot |\mu_{j,l}| \\ &\leq m_i + (m_i + \frac{1}{2})\frac{1}{2} = \frac{3}{2}m_i + \frac{1}{4} \leq 2m_i\end{aligned}$$

Thus the new value of m_i cannot be more than twice the old value. So during the whole loop, m_i increases by at most a factor $2^{i-1} \leq 2^{n-1}$, so m_i^2 increases by at most a factor 2^{2n-2}. Thus it is always the case that $m_i^2 \leq nA^n 2^{2n-2} \leq n(4A)^n$. This gives

$$q(b_i) \leq nm_i^2 A \leq n^2(4A)^n A \leq n^2(4A)^{n+1}.$$

Fourth step. It remains to bound the absolute values of the numerators of the numbers occurring in the algorithm:

1. $||b_k|| = q(b_k)^{\frac{1}{2}} \leq n(4A)^{\frac{n+1}{2}}$
2. $||D_{k-1}b_k^*|| \leq A^n A^{\frac{1}{2}}$
3. $|D_l \mu_{k,l}| \leq A^n (D_{l-1})^{\frac{1}{2}} ||b_k|| \leq A^n A^{\frac{n}{2}} n(4A)^{\frac{n+1}{2}} \leq n(4A)^{2n+\frac{1}{2}}$

Thus the numerators all have length $O(n \log A)$ also, completing the proof. □

Here is a theorem on a variation of the LLL-algorithm, using floating point arithmetic:

Theorem 2.46 (NGuyen-Stehlé). *Let $\Lambda \subset \mathbb{Z}^d$ be a lattice given by a generating family of n vectors. Then a reduced basis can be obtained in time*

$$O(d^4 n(d + \log A) \log A).$$

When $n = d$ this becomes $O(n^6 \log^2 A)$.

2.3 Applications

2.3.1 Simultaneous diophantine approximation

The problem is the following one. Some real numbers $x_1, \dots, x_d$ being given, find $p_1, \dots, p_d \in \mathbb{Z}$ and $q \in \mathbb{Z}_{>0}$ such that for all i, $|x_i - p_i/q|$ is small. Of course, there is a link between the size of q and $\max |x_i - p_i/q|$.

Theorem 2.47 (Dirichlet)**.** *Let $Q \in \mathbb{Z}_{>0}$. There exists $q \in \mathbb{Z}_{>0}$ such that*

$$0 < q \le Q^d \quad \textit{and} \quad \Big| qx_i - \lfloor qx_i \rceil \Big| < \frac{1}{Q} \textit{ for every } i.$$

Proof. For $q \in [1, Q^d + 1]$ let us put

$$s_q = \big(qx_1 \text{ cmod } 1, \dots, qx_d \text{ cmod } 1\big) = \big(qx_1 - \lfloor qx_1 \rceil, \dots, qx_d - \lfloor qx_d \rceil\big).$$

We have $s_q \in [-1/2, 1/2[^d$. Let us put

$$S = \Big\{ s_q;\ q \in [1, Q^d + 1] \Big\} \subset [-1/2, 1/2[^d.$$

It is a set of cardinality $Q^d + 1$, unless we have $x_i \in 1/t\mathbb{Z}$ for every i, where $t \in [1, Q^d]$. But in this case the result is trivial and we suppose from now on that we are not in this situation. Let us decompose $[-1/2, 1/2[^d$ in the Q^d parts $\prod [-1/2 + v_i/Q, -1/2 + (v_i + 1)/Q[$ where $v \in [0, Q-1]^d$. By the Dirichlet's principle, two elements of S are in the same part, say $(q_1x_1 \text{ cmod } 1, \dots, q_1x_d \text{ cmod } 1)$ and $(q_2x_1 \text{ cmod } 1, \dots, q_2x_d \text{ cmod } 1)$. For every i we have $|(q_1x_i \text{ cmod } 1) - (q_2x_i \text{ cmod } 1)| < 1/Q$. Taking $q = |q_1 - q_2|$ which is $\le Q^d$ gives the result. □

The problem is that this proof can be effective but is not efficient. Now, using the LLL-reduction algorithm we find a similar result.

Theorem 2.48. *Let Q be an integer $> 2^{d/4}$ and let $(x_1, \dots, x_d) \in \mathbb{R}^d$. There exists a deterministic polynomial running time algorithm which finds $q \in [1, 2^{d/4}Q^d]$ such that*

$$\max_i \Big| qx_i - \lfloor qx_i \rceil \Big| \le 3 \cdot 2^{(d-4)/4} \frac{1}{Q}.$$

Proof. Consider the lattice of $\mathbb{R}^{d+1}$ $(\Lambda, \| \cdot \|^2)$ generated by the columns of the matrix

$$B = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \lfloor Cx_1 \rceil & C & \ddots & 0 \\ \vdots & & \ddots & 0 \\ \lfloor Cx_d \rceil & 0 & \cdots & C \end{pmatrix}.$$

where C is a constant to be fixed later. According to Proposition 2.35, LLL finds $s \in \Lambda \setminus \{0\}$ such that

$$\| s \| \leq 2^{d/4} \mathrm{d}(\Lambda)^{1/(d+1)} = 2^{d/4} C^{d/(d+1)}.$$

Decompose s in the basis (b_i):

$$\begin{aligned} s &= qb_1 - n_1 b_2 - \cdots - n_d b_{d+1} \\ &= \Big(q, q\lfloor Cx_1 \rceil - n_1 C, \ldots, q\lfloor Cx_d \rceil - n_d C \Big). \end{aligned}$$

If $q \neq 0$, we can suppose $q > 0$ (if not replace s by $-s$). We can bound each of the coordinates of s by $\| s \|$ so that

$$q \leq \| s \| \quad \text{and} \quad \Big| q\lfloor Cx_i \rceil - n_i C \Big| \leq \| s \| \quad \text{for every } i \geq 1.$$

Let us show now that the q found by LLL is suitable. We have

$$\begin{aligned} \Big| qx_i - \lfloor qx_i \rceil \Big| &\leq |qx_i - n_i| \\ &\leq \left| \frac{q\lfloor Cx_i \rceil}{C} - n_i \right| + \left| \frac{q\lfloor Cx_i \rceil}{C} - qx_i \right| \\ &\leq \frac{1}{C} \Big(\big| q\lfloor Cx_i \rceil - n_i C \big| + \big| \lfloor Cx_i \rceil - Cx_i \big| \cdot q \Big) \\ &\leq \frac{1}{C} \Big(\| s \| + \frac{1}{2} \| s \| \Big) \\ &\leq \frac{3}{2C} \cdot 2^{d/4} C^{d/(d+1)} \\ &\leq 3 \cdot 2^{(d-4)/4} C^{-1/(d+1)}. \end{aligned}$$

It is sufficient to take $C = Q^{d+1}$ to have the inequalities of the Theorem:

$$q \leq 2^{d/4} Q^d \quad \text{and} \quad \Big| qx_i - \lfloor qx_i \rceil \Big| \leq 3 \cdot 2^{(d-4)/4} \frac{1}{Q} \quad \text{for every } i.$$

Now we must be sure that LLL does not return $q = 0$. Every non zero linear combination of the $(b_i)_{i \geq 2}$ has norm $\geq C$. Thus it is sufficient to fix a bound such that $\| s \| < C$.

$$\| s \| \leq 2^{d/4} \cdot C^{d/(d+1)} = 2^{d/4} \frac{C}{Q}.$$

If we take $Q > 2^{d/4}$, we have $\| s \| < C$ and then $q \neq 0$. □

2.3.2 Algebraicity test

Let $x \in \mathbb{R}$, given by a decimal approximation $\widehat{x_\varepsilon}$:

$$|x - \widehat{x_\varepsilon}| < \varepsilon.$$

(We think of $\widehat{x_\varepsilon} \in \mathbb{Q}$ as an approximation, which can be made arbitrarily precise.) We want to answer the question: is x algebraic? Of course, there is no way to prove this only by knowing $\widehat{x_\varepsilon} \in \mathbb{Q}$ for a given ε, since this rational number is obviously algebraic! But there is a nice way to make good guesses provided ε is sufficiently small compared to the height and degree of x, which measure the "complexity" of an algebraic number.

We fix a positive integer n, and a big real number $C > 0$; think of n as an upper bound for the degree of a minimal polynomial of x. Consider the $(n+2) \times (n+1)$ matrix

$$A = \begin{pmatrix} 1 & \dots & \dots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & \dots & 1 \\ Cx^n & \dots & Cx & C \end{pmatrix}$$

and the $\mathbb{Z}$-module $\subset (\mathbb{R}^{n+2}, \|\cdot\|)$ generated by the columns of A, that is the set of

$$\begin{pmatrix} \lambda_n \\ \vdots \\ \lambda_0 \\ C\left(\sum_{i=0}^n \lambda_i x^i\right) \end{pmatrix}, \quad \lambda_i \in \mathbb{Z}.$$

We are interested in short vectors in this lattice, where "short" means small with respect to the Euclidean length. A short vector satisfies:

1. $\sum_{i=0}^n \lambda_i^2$ is small;

2. $C^2 \left(\sum_{i=0}^n \lambda_i x^i\right)^2$ is small.

If C is large, then we probably have $\sum_{i=0}^n \lambda_i x^i \approx 0$ for a short vector, so we let

$$P(X) = \sum_{i=0}^n \lambda_i X^i.$$

If x is algebraic of degree $\leq n$ then we can hope that $P(x) = 0$. We shall see how to *guarantee* this below. The point of the first condition is that if we allow λ_i to be arbitrarily large, the pigeonhole principle says there are many good approximations satisfying the second one, possibly bearing no relation to the minimal polynomial we look for. Assuming x is decent, λ_i should be relatively small. Let us give now the results.

Theorem 2.49. *Let $z \in \mathbb{C}$. Assume*

1. *There exists $P \in \mathbb{Z}[X]$ with $P(z) = 0$, $mdegP \leq n$, and $\| P \|_\infty \leq A$.*
2. *We can compute $\hat{z}(\varepsilon) \in \mathbb{Q}(i)$ such that $|z - \hat{z}(\varepsilon)| < \varepsilon$, for all $\varepsilon > 0$ with $\log(1/\varepsilon) \leq f(A, n) = O(n^2 \log A)$ (where f is a rather complicated function to be defined in the proof).*

Then, if we are lucky, we can find such a P in deterministic polynomial time (actually, polynomial in $n \log A$).

Proof. We now prove the Theorem: Let Λ be the free $\mathbb{Z}$-module generated by the columns of the following matrix of $M_{(n+3)\times(n+1)}(\mathbb{R})$.

$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ C\mathrm{Re}(\hat{z}^n) & \dots & C\mathrm{Re}(\hat{z}^0) \\ C\mathrm{Im}(\hat{z}^n) & \dots & C\mathrm{Im}(\hat{z}^0) \end{pmatrix},$$

where $C > 1$ is a large integer. In other words, the set of

$$\begin{pmatrix} \lambda_n \\ \vdots \\ \lambda_0 \\ C\mathrm{Re}(\sum \lambda_i \hat{z}^i) \\ C\mathrm{Im}(\sum \lambda_i \hat{z}^i) \end{pmatrix}, \quad \lambda_i \in \mathbb{Z}.$$

Here $\Lambda \subset \mathbb{R}^{n+3}$ has rank $n+1$ so that *we are not exactly in our initial context*, namely a basis of the whole space. It is possible however to check that our following arguments are still correct.

Consider now the lattice (Λ, q) where q is the Euclidean form of $\mathbb{R}^{n+3}$ and apply LLL.

Let $v = (\lambda_n, \dots, \lambda_0, *, *) \in \Lambda$ be the first vector in the obtained reduced basis, and let $Q(X) = \sum_{i=0}^n \lambda_i X^i$. Assume that $\| Q \|_\infty \leq A$ (we are lucky). We claim that $Q(z) = 0$, provided C was large enough! To prove this, we need three lemmas:

Lemma 2.50 (Cauchy's bound). *Let $z \in \mathbb{C}$, $P = \sum_{i=0}^{n} a_i X^i \in \mathbb{C}[X]$ such that $P(z) = 0$, $a_n \neq 0$. Then*

$$|z| \leq 2 \max_{0 \leq i < n} \left| \frac{a_i}{a_n} \right|^{1/(n-i)}.$$

Proof. Assume by contradiction that

$$|z|^{n-i} > 2^{n-i} \left| \frac{a_i}{a_n} \right|, \quad \text{for all } i < n.$$

Then $2^{i-n}|a_n||z|^n > |a_i z^i|$ and

$$|a_n z^n| > |a_n z^n| \sum_{i<n} 2^{i-n} > \sum_{i<n} |a_i z^i|.$$

Hence $P(z) \neq 0$. □

Lemma 2.51. *Let $P \in \mathbb{Z}[X]$, irreducible over $\mathbb{Q}$, $mdegP \leq n$, $\| P \|_\infty \leq A$, and $z \in \mathbb{C}$ a root of P. Then for all $Q \in \mathbb{Z}[X]$ such that $mdegQ \leq n$ and $\| Q \|_\infty \leq A$, we have either $Q(z) = 0$ or $|Q(z)| \geq \eta(A, n) > 0$.*

Proof. We may write $P = a_n \prod_{i=1}^{n} (X - z_i)$, $z_i \in \mathbb{C}$, with $z_1 = z$ say. From Lemma 2.50, $|z_i| \leq D = 2A$ and we may assume $D \geq 1$. Now consider

$$R := a_n^{mdegQ} \prod_{i=1}^{n} Q(z_i) = \mathrm{Res}(P, Q) \in \mathbb{Z}.$$

Recall that $\mathrm{Res}(P, Q) = 0$ if and only if P and Q have a common root in $\mathbb{C}$, which implies $P \mid Q$ since P is irreducible. Assume that $Q(z) \neq 0$, then P and Q have no common root and $|\mathrm{Res}(P, Q)| \geq 1$.

On the other hand, we bound $|Q(z_i)| \leq D^n A(n+1)$ for all $i > 1$, $|a_n|$ by A, and obtain

$$1 \leq |R| \leq Q(z) A^n \big(D^n A(n+1)\big)^{n-1},$$

so that the result holds with

$$\eta(A, n) = \frac{1}{2^{n(n-1)} A^{n^2+n-1} (n+1)^{n-1}}.$$

Note that $\log(1/\eta) = O(n^2 \log A)$. □

Lemma 2.52. *Let $z, \hat{z} \in \mathbb{C}$ such that $|z - \hat{z}| < \varepsilon < 1$, $|z| \leq 2A$, and let $P \in \mathbb{C}[X]$, $mdegP \leq n$, $\| P \|_\infty \leq A$. Then $|P(z) - P(\hat{z})| < \varepsilon B(A, n)$.*

Proof. From the Mean Value Theorem, $P(z) - P(\hat{z}) = (z - \hat{z})P'(\xi)$, $\xi = z + t(\hat{z} - z)$, for some $t \in [0, 1]$. Hence $|\xi| \leq 2A + \varepsilon \leq 2A + 1$ and $|P'(\xi)| \leq n^2A(2A + 1)^{n-1}$, so that we have our result with

$$B(A, n) = n^2A(2A + 1)^{n-1}.$$

Note that $\log B = O(n \log A)$. □

We continue with the proof of theorem. Recall that C and ε are not fixed, we will choose them in order to get a contradiction.

- Assume by contradiction that $Q(z) \neq 0$. then $C|Q(z)| > C\eta(A, n)$ by Lemma 2.51. Hence, $C|Q(\hat{z}| > C(\eta - \varepsilon B)$ (Lemma 2.52). Note that the statement is empty if $\varepsilon B > \eta$.
- Since v is the first vector in a reduced basis, the LLL theorem says that

$$\begin{array}{rcll} q(v) & \leq & 2^n q & \text{(smallest vector in } \Lambda \text{ – see proof of Theorem 2.37)} \\ & \leq & 2^n\big(C^2|P(\hat{z})|^2 + (n+1)A^2\big) & \text{(take } \lambda_i \text{ as coeff. of } P) \\ & \leq & 2^n\big(C^2\varepsilon^2B^2 + (n+1)A^2\big) & \text{(Lemma 2.52, with } P(z) = 0). \end{array}$$

- On the other hand, by definition of v, $q(v) \geq C^2|Q(\hat{z})|^2 > C^2(\eta - \varepsilon B)^2$ if $\eta - \varepsilon B \geq 0$.

We get a contradiction if

$$\begin{array}{rcl} C^2(\eta - \varepsilon B)^2 & \geq & 2^n(C^2\varepsilon^2B^2 + (n+1)A^2), \\ \text{or} \quad \eta - \varepsilon B & \geq & 2^{(n+1)/2}\varepsilon B, \end{array}$$

if we choose C such that $C\varepsilon B = \sqrt{n+1}A$. We now choose $\varepsilon \leq \eta/2B$ such that $2^{(n+1)/2}\varepsilon B \leq \eta/2$, i.e. $\varepsilon = \eta\big/B2^{(n+3)/2}$ is suitable. Finally, our function f can be taken as

$$f(A, n) = 2\log n + (n^2 - n + 1)\log 2 + (n^2 + n)\log A + (n-1)\log((A+1)(n+1)).$$

Funny is'nt it! We find successively

$$\log(1/\varepsilon) = \log(1/\eta) + \log B + O(n) = O(n^2 \log A),$$

$$\log C = O(\log(nA) - \log(\varepsilon B)) = O(n^2 \log A),$$

hence the sizes of all inputs are polynomially bounded in terms of $n \log(A)$. □

Chapter 3

Polynomials

3.1 Factoring in $\mathbb{F}_q[X]$

3.1.1 Finite fields

Let us first summarize some well known facts about finite fields.

- Let F be a finite field. The additive order of 1_F is a prime number p, which is also the characteristic of F. The subset $\{0, 1_F, \ldots, (p-1)\cdot 1_F\}$ is the smallest subfield of F and is isomorphic to $\mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p$.

- The field F can be viewed as a finite dimensional vector space over $\mathbb{F}_p$ so that there exists $f \geq 1$ such that $\sharp F = p^f$.

- For every prime p and every $f \geq 1$, there exists a (unique up to non canonical isomorphism) field of cardinality $q = p^f$ (take an irreductible polynomial P of $\mathbb{F}_p[X]$ of degree f – such a polynomial exists – and consider $\mathbb{F}_p[X]/(P)$).

- By Lagrange Theorem For every $x \in \mathbb{F}_q^\times$ we have $x^{q-1} = 1$ and

$$X^q - X = \prod_{\alpha \in \mathbb{F}_q} (X - \alpha). \tag{3.1}$$

- If $a_i \in \mathbb{F}_q$ for $0 \leq i \leq d$, we have

$$\Big(\sum_{i=0}^{d} a_i X^i\Big)^p = \sum_{i=0}^{d} a_i^p X^{ip}. \tag{3.2}$$

- $\mathbb{F}_q^\times$ is cyclic, isomorphic to $\mathbb{Z}/(q-1)\mathbb{Z}$ (the reason why, is that in every commutative group G there is an element whose order is the lcm s of the orders of the elements of G; here for every $x \in \mathbb{F}_q^\times$ we have $x^s = 1$ so that $s \geq q-1$; but $s \mid q-1$ and we have equality).

- The function $x \mapsto x^p$ from $\mathbb{F}_q$ to $\mathbb{F}_q$ where $q = p^f$ is an automorphism of $\mathbb{F}_q$ called the *Frobenius* automorphisme of $\mathbb{F}_q$. The group of automorphisms of $\mathbb{F}_q$ is cyclic, generated by $x \mapsto x^p$, and has f elements.

- Suppose that q is odd. In $\mathbb{F}_q$ there are $(q-1)/2$ quadratic residues i.e. elements which are squares of non zero elements. And x is a quadratic residue iff $x^{(q-1)/2} = 1$.

Our goal is to find a way to factor $P \in \mathbb{F}_q[X]$, that we can take monic, into a product of irreductible polynomials.

3.1.2 Squarefree factorization

We now reduce the general problem of factorization over $k[X]$ to *squarefree* polynomials. Recall that P is squarefree if there is no Q with degree > 0 such that $Q^2 \mid P$.

Definition 3.1. Let $T \in k[X]$ be monic non-constant, and $\prod T_i^{e_i}$ its decomposition into distinct irreducible monic factor. The *squarefree part* of T, or $\mathrm{core}(T)$, is $\prod T_i$. The *squarefree factorization* of T is $T = f_1^1 f_2^2 \dots f_m^m$, where the $f_i \in k[X]$ are monic squarefree polynomials, pairwise coprime, and $f_m \neq 1$. The squarefree part of T is $f_1 \dots f_m$.

It is enough for our purpose to compute $\mathrm{core}(T)$, since once it is factored and the irreducible factors of T are known, we can compute the e_i as valuations (if they are at all needed) in the following way: compute $T/\mathrm{core}(T)$ and its squarefree part, then iterate. It is an effective but naive approach which can be improved (see exercise on Yun' algorithm).

Lemma 3.2. *Let $T \in k[X]$ of degree n, k a field of characteristic $p \geq 0$. Assume $T = \prod_{i=1}^s T_i^{e_i}$, where the T_i are irreducible, pairwise non-associate. Let*

$$u = \gcd(T, T'), \quad v = T/u, \quad w = \frac{u}{\gcd(u, v^{n-1})}$$

then

$$v = \prod_{p \nmid e_i} T_i, \quad \textit{and} \quad w = \prod_{p \mid e_i} T_i^{e_i}.$$

Proof. From the logarithmic derivative formula, we find

$$T' = \sum_{i=1}^{s} e_i \frac{T_i'}{T_i} T,$$

hence $T_i^{e_i-1} \mid \gcd(T, T')$ and $T_i^{e_i}$ divides the GCD if and only if $T_i \mid e_i T_i'$, i.e $e_i = 0$ in k. Since the irreducible factors of u are among the T_i,

$$u = \prod_{p \nmid e_i} T_i^{e_i-1} \prod_{p \mid e_i} T_i^{e_i},$$

and the first equality follows. For the second, we have $e_i - 1 \leq n - 1$ for all i, hence

$$\gcd(u, v^{n-1}) = \prod_{p \nmid e_i} T_i^{e_i-1}$$

and we are done. □

From the lemma, we can compute the squarefree part of T, in fact in a partially factored form.

Algorithm 5. Squarefree part over $\mathbb{F}_q$, core

Input: $T \in \mathbb{F}_q[X]$.

Output: The squarefree part of T, $\mathrm{core}(T)$.

```
1: Compute $u = \gcd(T, T')$, then $v = T/u$.
2: Compute a representative $a \in \mathbb{F}_q[X]$ of $v^{n-1}$ in $\mathbb{F}_q[X]/(u)$, then $w = u/\gcd(u, a)$.   {$w$ is of the form $\sum w_i X^{pi}$ – see (3.2)}
3: Let $W = \sum F^{-1}(w_i) X^i$, where $F^{-1} : x \mapsto x^{q/p}$.   {we have $W^p = w$ – see (3.1) and (3.2)}
4: Return $v \times \mathrm{core}(W)$, calling ourselves recursively.
```

Theorem 3.3. *Let $T \in k[X]$ be of degree n. If k has characteristic 0, or $k = \mathbb{F}_p$, p prime, the squarefree part of T is computed in $\widetilde{O}(n)$ operations in k. If $k = \mathbb{F}_q$ is finite but not a prime field, the squarefree part is computed in $\widetilde{O}(n \log q)$ operations.*

Proof. Computing u, v, $v^{n-1} \bmod u$ and w all cost $\widetilde{O}(n)$ operations in k. This proves the case $\mathrm{char} k = 0$. Evaluating F^{-1} on $x \in \mathbb{F}_q$ costs $\widetilde{O}(\log(q/p))$ operations, i.e. nothing if $q = p$. If $f(n)$ is the total cost of the computation for T of degree n, then $f(n) \leq \widetilde{O}(n \log(q/p)) + f(n/p)$, and the case $k = \mathbb{F}_q$ follows from a variant of Lemma 1.8. □

Remark 3.4. If we are counting operations over $\mathbb{F}_p$, and write $\mathbb{F}_q = \mathbb{F}_p(y)$ then we can improve the above by precomputing $F^{-1}(y)$ then $F^{-1}(Q(y))$ as $Q(F^{-1}(y))$.

Yun's algorithm exploits the Lemma in a clever way to find the full squarefree factorization at the same cost as above (see exercise 2).

3.1.3 Berlekamp algorithm for small q

Let $T \in \mathbb{F}_q[X]$ be squarefree, $T = \prod_{i=1}^s T_i$, T_i irreducible, pairwise coprime. Let $A = \mathbb{F}_q[X]/(T)$ and $F : x \mapsto x^q$ the Frobenius endomorphism (of $\mathbb{F}_q$-algebras) of A. By the Chinese Remainder Theorem,

$$\begin{array}{rcl} A & \overset{\sim}{\longrightarrow} & \prod_{i=1}^{s} \mathbb{F}_q[X]/(T_i) \simeq \prod_{i=1}^{s} \mathbb{F}_{p^{\deg T_i}}. \\ \bar{a} & \longmapsto & (\pi_1(a), ..., \pi_s(a)) \end{array}$$

where $a \in \mathbb{F}_q[X]$ and $\pi_i(a) = a \bmod T_i$. Let

$$B = \ker(F - \mathrm{Id}) = \{\bar{b} \in A;\ b \in \mathbb{F}_q[X],\ b^q - b \equiv 0 \bmod T\},$$

the *Berlekamp* algebra.

Lemma 3.5. $B \simeq (\mathbb{F}_q)^s$.

Proof. In each field $\mathbb{F}_q[X]/(T_i)$, the equation $x^q = x$ has exactly q solutions, the elements of $\mathbb{F}_q$. This implies that there are q^s polynomials Q modulo T which are solutions of $Q^q \equiv Q \bmod T$ and each solution corresponds to one of the q^s vectors $(\alpha_1, \dots, \alpha_s) \in \mathbb{F}_q^s$ by $Q \equiv \alpha_i \bmod T_i$ for $i = 1, \dots, s$. □

Lemma 3.6. *If $b \in \mathbb{F}_q[X]$ is such that $\bar{b} \in B$, we have*

$$T = \prod_{u \in \mathbb{F}_q} \gcd(T, b-u).$$

Proof. We have $X^q - X = \prod_{u\in\mathbb{F}_q}(X-u)$ so that $b^q - b = \prod_{u\in\mathbb{F}_q}(b-u)$. Now T divides $b^q - b$ so that $\gcd(T, b^q - b) = T$. Therefore

$$T = \gcd\Big(T, \prod_{u\in\mathbb{F}_q}(b-u)\Big) = \prod_{u\in\mathbb{F}_q} \gcd(T, b-u),$$

where the second equality follows from the fact that if $\gcd(b,c) = 1$ then $\gcd(a,bc)$ =$\gcd(a,b)\gcd(a,c)$. □

This leads to the following algorithm.

```
Algorithm 6. Berlekamp algorithm
Input: $T \in \mathbb{F}_q[X]$ squarefree.
Output: The irreducible factors of $T$.
Compute an $\mathbb{F}_q$-basis of $B = \ker(F - \mathrm{Id})$ acting on $\mathbb{F}_q[X]/(T)$. Denote
  it by $(\bar{b_1}, \ldots, \bar{b_s})$ where $s = \dim_{\mathbb{F}_q} B$.
if $s = 1$ then
    we are done: $T$ is irreducible
else
    set $\mathcal{F} = \{T\}$ and for $i$ from 1 to $s$ replace each $h \in \mathcal{F}$ by the nontrivial
      elements of $\{\gcd(h, b_i - u);\ u \in \mathbb{F}_q\}$.
Return $\mathcal{F}$.
```

Theorem 3.7. *The Berlekamp algorithm terminates, gives the desired decomposition of T. The total number of operations in $\mathbb{F}_q$ is in $\widetilde{O}(n^\omega + n^2 q)$, where n is the degree of T.*

Proof. The correctness of the algorithm follows from previous lemmas. We note that throughout the algorithm T is equal to the product of all elements of $\mathcal{F}$. This follows immediately from Lemma 3.6 applied to h (instead of T) because $T \mid b^q - b \Longrightarrow h \mid b^q - b$ and indeed we have $h = \prod_{u \in \mathbb{F}_q} \gcd(h, b_i - u)$ for every i. Also the elements of $\mathcal{F}$ are pairewise relatively prime, as $b_i - u$ and $b_i - v$ are relatively prime for $u \neq v$, and by induction. So the only thing that could be wrong with the output is that it contains an element which is divided at least by two distinct T_i.
Let h be an element returned by the algorithm. Then for each $1 \leq i \leq s$ there is an $u_i \in \mathbb{F}_q$ such that $b_i \equiv u_i \bmod h$: for a fixed i this is true after the execution of splitting thanks to b_i and it remains true as in subsequent steps a polynomial is replaced by factors of it. Let $\bar{b} \in B$. Then there are $\beta_i \in \mathbb{F}_q$ with $b = \sum_{i=1}^s \beta_i b_i$. Hence if we set $u = \sum_{i=1}^s \beta_i u_i$ we have $b \equiv u \bmod h$. Now suppose that h contains two distinct T_i, say T_1 and T_2. Then for $\bar{b} \in B$ we have $\pi_1(b) = \pi_2(b) = u$, but this contradicts the isomorphism $B \simeq (\mathbb{F}_q)^s$.

Now, what is the cost of computing a basis of B? We count operations over $\mathbb{F}_q$:

1. Compute $F(X) = X^q$ in $A = \mathbb{F}_q[X]/(T)$: $\widetilde{O}(n \log q)$ operations.

2. $F(X^2) = X^{2q}, \ldots, X^{(n-1)q}$: $n - 2$ multiplications in A, for a cost of $\widetilde{O}(n^2)$ operations ($\widetilde{O}(n)$ for one multiplication).

3. Compute the kernel : $\widetilde{O}(n^\omega)$.

For each b_i, the gcd computation is in $\widetilde{O}(n)$ and we make it q times. Then this is done $s \le n$ times. In fact, it is possible to prove that on average, s is approximately $\log n$ (see Knuth for instance). □

If the algorithm is fast for small q, it is too slow for large q. In the two next sections we see how to improve our algorithm using a probabilistic approach.

3.1.4 Berlekamp algorithm for large q odd

From now on $q = p^f$ is odd. Let us note that for all $\bar{b} \in B$,

$$\bar{b}(\bar{b}^t - 1)(\bar{b}^t + 1) = 0,$$

where $t = \frac{q-1}{2}$. Let $\bar{b} \in B$. Suppose T is reducible i.e. $s \ge 2$. The nontrivial factors common to T and $b^q - b$ are spread out amongst b, $b^t - 1$ and $b^t + 1$ and it is reasonable to expect that almost half of them are either factors of $b^t - 1$ or $b^t + 1$. In fact we have the following result.

Lemma 3.8. *The element $\bar{b} \in B$ being chosen at random, the probability of* $\gcd(b^t - 1, T)$ *being nontrivial is*

$$1 - \Big(\frac{q-1}{2q}\Big)^s - \Big(\frac{q+1}{2q}\Big)^s \ge \frac{4}{9}.$$

Proof. Let $\bar{b} \simeq (\alpha_1, \dots, \alpha_s)$ where we have $\alpha_i \in \mathbb{F}_q$ for every i (see above). We have

$$\begin{aligned} T_i \mid \gcd(b^t - 1, T) &\iff \alpha_i^t = 1 \\ &\iff \alpha^{(q-1)/2} = 1 \\ &\iff \alpha_i \text{ quadratic residue mod } q. \end{aligned}$$

If we denote by $B_i \simeq \mathbb{F}_q$ the i-th composant of B we see that in each B_i we have exactly $(q-1)/2$ quadratic residues so that the probability to have $T_i \mid \gcd(b^t - 1, T)$ for every i (or $\gcd(b^t - 1, T) = T$) is

$$\Big(\frac{q-1}{2q}\Big)^s.$$

Similarly there are $(q+1)/2$ non quadratic residues modulo q and hence the probability to have $\gcd(b^t - 1, T) = 1$ is

$$\Big(\frac{q+1}{2q}\Big)^s,$$

so that finally the probability to have a nontrivial gcd is

$$\begin{aligned} 1-\left(\frac{q-1}{2q}\right)^s-\left(\frac{q+1}{2q}\right)^s &\geq 1-\left(\frac{q-1}{2q}\right)^2-\left(\frac{q+1}{2q}\right)^2 \\ &\geq \frac{1}{2}-\frac{1}{2q^2} \\ &\geq \frac{4}{9} \end{aligned}$$

since $s \geq 2$ and $q \geq 3$. □

This result leads to the following probabilistic algorithm.

Algorithm 7. SPLIT over $\mathbb{F}_q[X]$, q odd

```
Input: $(T, B, \mathcal{F})$, where $T \in \mathbb{F}_q[X]$, $B$ is given by an $\mathbb{F}_q$-basis, and $\mathcal{F} = \{f_1, \dots f_r\}$ is a collection of $f_i \in \mathbb{F}_q[X]$ such that $T = \prod_{i \leq r} f_i$ for some $r < s$.
Output: A larger collection of factors of $T$.
Pick uniformly a random $\overline{x} \in B$
Compute $\bar{y} := \overline{x}^t$ in $\mathbb{F}_q[X]/(T)$.    {$O(\log t)$ multiplications}
Compute $D := \gcd(y-1, f_i)$ in $\mathbb{F}_q[X]$, for all $i \leq r$. Each time $D$ is non trivial, we split the corresponding $f_i$ in two.
Return the new collection of factors.
```

The global algorithm is then as follows:

Algorithm 8. Berlekamp's algorithm over $\mathbb{F}_q[X]$, q odd

```
Input: $T \in \mathbb{F}_q[X]$, squarefree, $q$ odd.
Output: The irreducible factors of $T$.
Compute an $\mathbb{F}_q$-basis of $B = \ker(F - \mathrm{Id})$ acting on $\mathbb{F}_q[X]/(T)$.
Let $s = \dim_{\mathbb{F}_q} B$. If $s = 1$, we are done: $T$ is irreducible.
Otherwise, set $\mathcal{F} = \{T\}$ then $\mathcal{F} = \mathrm{SPLIT}(T, B, \mathcal{F})$ until $\sharp\mathcal{F} = s$.
Return $\mathcal{F}$.
```

Theorem 3.9. *The expected cost of "Splitting B", i.e. of splitting T using B is $\widetilde{O}(n \log q)$ operations in $\mathbb{F}_q$. The total cost of randomized Berlekamp algorithm for q odd is $\widetilde{O}(n^\omega + n \log q)$ expected operations in $\mathbb{F}_q$. The computation of the Berlekamp algebra B is deterministic, and yields the number of irreducible factors; only the splitting is randomized.*

Proof. Exercise. □

3.1.5 Berlekamp algorithm for large q even

As before, let $B = \ker(F - \mathrm{Id})$ be the Berlekamp algebra. For q odd, we used $x(x^t - 1)(x^t + 1) = 0$. In characteristic 2, i.e. when $q = 2^f$ we consider instead

$$\mathrm{Tr}(x) = \sum_{k=1}^{f} x^{2^{k-1}}.$$

Lemma 3.10. *The map* Tr *has the following properties.*

1. $\mathrm{Tr} : \mathbb{F}_{2^f} \to \mathbb{F}_2$ *is a surjective* $\mathbb{F}_2$*-linear map.*
2. *For a random* $x \in \mathbb{F}_{2^f}$, $\mathrm{Tr}(x) = 0$ *or* 1 *with probability* $1/2$.
3. *For all* $\bar{b} \in B$, $\pi_i(\mathrm{Tr}(b)) \in \mathbb{F}_2$.

Proof. The $\mathbb{F}_2$-linearity is trivial and comes from $\lambda^2 = \lambda$ if $\lambda \in \mathbb{F}_2$ and from the fact that $\mathbb{F}_{2^f}$ has characteristic 2. Now it is easy to see that

$$\begin{aligned}\mathrm{Tr}(x)(\mathrm{Tr}(x)+1) &= \mathrm{Tr}(x)^2 + \mathrm{Tr}(x)\\ &= \mathrm{Tr}(x^2) + \mathrm{Tr}(x)\\ &= x^{2^f} + x\\ &= 0\end{aligned}$$

for every $x \in \mathbb{F}_{2^f}$, which proves that $\mathrm{Tr}(x) = 0$ or 1. It follows from a degree consideration that as polynomials in $\mathbb{F}_{2^f}[X]$ we have

$$\mathrm{Tr}(X)(\mathrm{Tr}(X)+1) = X^{2^f} + X = \prod_{\alpha \in \mathbb{F}_{2^f}} (X - \alpha).$$

Each polynomial $\mathrm{Tr}(X)$ and $\mathrm{Tr}(X)+1$ has degree 2^{f-1} and has exactly 2^{f-1} roots in $\mathbb{F}_{2^f}$. This leads to the surjectivity of Tr and to point 2.

Finally we know that $B \simeq (\mathbb{F}_{2^f})^s$ and if $\bar{b} \in B$ we have $\pi_i(\mathrm{Tr}(b) = \mathrm{Tr}(\pi_i(b)) \in \mathbb{F}_2$ by point 1. □

Corollary 3.11. *If* $s \geq 2$*, for a random* $\bar{b} \in B$*, the probability of* $\gcd(T, \mathrm{Tr}(b))$ *being nontrivial is* $\geq 1/2$.

Proof. It is easy to see that $T_i \mid \gcd(T, \mathrm{Tr}(b)) \iff \pi_i(\mathrm{Tr}(b)) = 0$ so that by the previous lemma,

$$\gcd(T, \mathrm{Tr}(b)) \text{ is trivial } \iff \pi_i(\mathrm{Tr}(b)) = 0 \text{ or } \pi_i(\mathrm{Tr}(b)) = 1 \text{ for every } i.$$

Applying the lemma shows that this happens with probability $2 \cdot 2^{-s} \leq 1/2$. □

We thus obtain our last variant of the Split algorithm:

Algorithm 9. Split over $\mathbb{F}_q[X]$, $q = 2^f$ even

Input: $(T, B, \mathcal{F})$, where $T \in \mathbb{F}_q[X]$, B is given by an $\mathbb{F}_q$-basis, and $\mathcal{F} = \{f_1, \dots f_r\}$ is a collection of $f_i \in \mathbb{F}_q[X]$ such that $T = \prod_{i \le r} f_i$ for some $r < s$.

Output: A larger collection of factors of T.

```
Pick uniformly a random $\bar{b} \in B$
Compute $D := \gcd(\mathrm{Tr}(b), T)$ in $\mathbb{F}_q[X]$, for all $i \le r$. Each time $D$ is non trivial, we split the corresponding $f_i$ in two.
Return the new collection of factors.
```

Theorem 3.12. *The expected cost of the randomized Berlekamp algorithm for q even is the same as in the odd case: $\widetilde{O}(n^\omega + n \log q)$ operations in $\mathbb{F}_q$.*

Proof. Exercise. □

3.1.6 Conclusion

Using black-box algebra and randomizing also the computation of the Berlekamp algebra B, the above improves to $\widetilde{O}(n^2 + n \log q)$ (Kaltofen & Lobo).

A quite different approach (iterated Frobenius), was introduced by von zur Gathen & Shoup, improving on a classical algorithm by Cantor & Zassenhaus: it avoids completely linear algebra, and computes simultaneously the X^{q^d} for $d \le n$ in $\widetilde{O}(n^2 + n \log q)$ operations in $\mathbb{F}_q$ (this is non-trivial within that time bound, contrary to the X^{qd} required by Berlekamp!).

After taking the gcds $(X^{q^d} - X, T)$, it remains to split products of irreducible polynomials having the same degree d. This algorithm achieves the same complexity: $\widetilde{O}(n^2 + n \log q)$ expected operations in $\mathbb{F}_q$. Both algorithms are quite practical.

3.2 Factoring in $\mathbb{Q}[X]$

3.2.1 Preliminaries

Let R be a Unique Factorization Domain (essentially $R = \mathbb{Z}$ or $F[Y]$ for a field F). The *content* $\mathrm{cont}(f)$ of a polynomial $f \in R[X]$ is the gcd of its coefficients (with the convention that it is positive if $R = \mathbb{Z}$ and monic if $R = F[Y]$). The *primitive part* of f is $\mathrm{pp}(f) = f/\mathrm{cont}(f)$. Gauss' Lemma says that for any f, $g \in R[X]$ we have

$$\mathrm{cont}(fg) = \mathrm{cont}(f)\mathrm{cont}(g) \quad \text{and} \quad \mathrm{pp}(fg) = \mathrm{pp}(f)\mathrm{pp}(g).$$

As a consequence

- $R[X]$ is also a UFD;
- its primes are the primes of R and the primitive polynomials ($\text{cont}(f) = 1$) in $R[X]$ that are irreductible in $K[X]$, where K is the field of fractions of R.

Thus the relation between factoring polynomials over $\mathbb{Q}$ or $\mathbb{Z}$ is as follows.

If $f \in \mathbb{Z}[X]$ is primitive, a factorization $f = f_1 \cdots f_k$ into irreducible $f_i \in \mathbb{Q}[X]$ yields a factorization $f = f_1^* \cdots f_k^*$ into irreducible $f_i^* \in \mathbb{Z}[X]$ by multiplying up denominators and then removing contents. In particular if $f \in \mathbb{Z}[X]$ is monic and if $f = f_1 \cdots f_s$ is a factorization of f in $\mathbb{Q}[X]$ where the f_i are monic, then the f_i are in fact in $\mathbb{Z}[X]$ and we have a factorization in $\mathbb{Z}[X]$.

On the other hand, any irreducible factorization in $\mathbb{Z}[X]$ is also one in $\mathbb{Q}[X]$. For any arbitrary f the factorization of f is the factorization of the content of f together with the factorization of its primitive part. Thus

$$\text{Factoring in } \mathbb{Z}[X] \Longleftrightarrow \text{ Factoring in } \mathbb{Q}[X] \text{ plus factoring in } \mathbb{Z}.$$

Unfortunately the best known algorithms for factoring in $\mathbb{Z}$ are much less efficient than those for $\mathbb{Q}[X]$, so that factoring in $\mathbb{Q}[X]$ is easier than in $\mathbb{Z}[X]$ even if it provides the factorization of the primitive part. However our goal here is to factor in $\mathbb{Q}[X]$.
Consider a polynomial $f = \sum_{i=0}^n a_i X^i \in \mathbb{Q}[X]$ that we want to factor. Cleaning denominators, we are reduced to factor a polynomial $g = \sum_{i=0}^n b_i X^i \in \mathbb{Z}[X]$ ($g = f \times \text{lcm}(d_i)$ where $a_i = c_i/d_i$). Put

$$h(X) = b_n^{n-1} g\Big(\frac{X}{b_n}\Big).$$

We have $h \in \mathbb{Z}[X]$, h monic and a factorization of h gives us a factorization of g (and of f) because $g(X) = b_n^{1-n} h(b_n X)$. So we are reduced to factor a monic polynomial of $\mathbb{Z}[X]$ and by what we have already seen we know that the monic irreducible factor of h in $\mathbb{Q}[X]$ are in fact in $\mathbb{Z}[X]$ so that we have to factor over $\mathbb{Z}$. Moreover, from what has been seen in section 3.1.2, we can assume that the monic polynomial of $\mathbb{Z}[X]$ that we want to factor is squarefree. In the general case, Yun's algorithm (see exercise 2 of Chapter 3) will give us the full squarefree factorization $h_1 h_2^2 \cdots h_m^m$ and factoring $h_1 \cdots h_m$ is sufficient. Finally we have to factor over $\mathbb{Z}$ a monic squarefree polynomial of $\mathbb{Z}[X]$.

3.2.2 Bounds

Let

$$T = a_n \prod_{i=1}^{n} (X - \alpha_i), \quad \alpha_i \in \mathbb{C}, \quad a_n \neq 0.$$

Define the Mahler measure

$$M(T) = |a_n| \prod_{i=1}^{n} \max(|\alpha_i|, 1).$$

It is an easy exercise to check that M is multipicative, i.e. $M(PQ) = M(P)M(Q)$ and that $M(X^n P(1/X)) = M(P)$ if n =deg$P$.

Lemma 3.13. *If $z \in \mathbb{C}$, $\| (X - z)T \|_2 = \| (\bar{z}X - 1)T \|_2$.*

Proof. In fact,

$$\| P \|_2^2 = \frac{1}{2\pi} \int_0^{2\pi} |P(e^{it})|^2 dt,$$

and the result follows from $|e^{it} - z| = |e^{-it} - \bar{z}| = |1 - \bar{z}e^{it}|$. □

Theorem 3.14 (Landau). *$M(T) \leq \| T \|_2 \leq 2^{\deg T} M(T)$.*

Proof. We first prove the left-hand side: let

$$U(X) = a_n \prod_{|\alpha_i| \leq 1} (\bar{\alpha}_i X - 1) \prod_{|\alpha_i| > 1} (X - \alpha_i) \in \mathbb{C}[X].$$

We have

$$M(T) = |U(0)| \leq \| U \|_2 = \| T \|_2,$$

where the last equality uses an obvious iteration of the previous lemma.

As for the right hand side, writing the relations between roots and coefficients for $T = \sum_{i=0}^{n} a_i X^i$, we have

$$|a_{n-i}| \leq \binom{n}{i} |a_n| \max_{j_1 < \cdots < j_i} |\alpha_{j_1} \cdots \alpha_{j_i}|,$$

and the max is less than $M(T)/|a_n|$ by definition of $M(T)$.

Then

$$\sum_{i=0}^{n} |a_i| \leq \sum_{i=0}^{n} \binom{n}{i} M(T) \leq 2^n M(T)$$

and the inequality follows from $\sum |a_i|^2 \leq (\sum |a_i|)^2$. □

Corollary 3.15 (Mignotte's bound)**.** *If $S \mid T$ in $\mathbb{Z}[X]$ then*

$$\| S \|_2 \leq 2^{\deg S} \| T \|_2 \leq 2^{\deg T} \| T \|_2 .$$

Proof. Since our polynomials are in $\mathbb{Z}[X]$ it is easy to see that $S \mid T$ implies $M(S) \leq M(T)$ so that

$$\| S \|_2 \leq 2^{\deg S} M(S) \leq 2^{\deg S} M(T) \leq 2^{\deg S} \| T \|_2 .$$

□

In particular, if $S \mid T \in \mathbb{Z}[X]$ and $\deg T = n$, we have

$$\| S \|_\infty \leq \| S \|_2 \leq 2^n (n+1)^{1/2} \| T \|_\infty,$$

yielding a finite number of possible divisors, hence a naive algorithm to factor T. We shall now improve drastically on such an exhaustive search.

3.2.3 Hensel lifting

First, a simple and general idea could be the following one. Suppose that we want to factor a monic squarefree polynomial $f \in \mathbb{Z}[X]$. Choose a prime p which does not divide $\mathrm{Res}(f, f')$. In this case, $\mathrm{Res}(f, f') \neq 0 \bmod p$ so that, if $\bar{g}$ is the notation for $g \bmod p$, $\bar{f}$ and $\bar{f}'$ are coprime and $\bar{f}$ is squarefree. So we can factor $\bar{f}$ in $\mathbb{F}_p[X]$ in an efficient way. Admit that $f = f_1 \cdots f_s$ is the factorization of f into monic irreducible polynomials in $\mathbb{Z}[X]$. Then

$$\bar{f} = \bar{f}_1 \cdots \bar{f}_s = g_1 \cdots g_r,$$

where $g_1, \ldots, g_r$ are the monic irreducible factors of $\bar{f}$ in $\mathbb{F}_p[X]$. Let for instance $S \subseteq \{1, \ldots, r\}$ be the set of indices of the irreducible factors of $\bar{f}_1$ (which is unknown). Then

$$f_1 \equiv \prod_{i \in S} g_i \bmod p.$$

If p is large enough in terms of the Mignotte bound, then the coefficients of f_1 are integers less than say $p/2$ and the upper equivalence is an equality if we use symmetric representatives. Therefore we can construct f_1 from the g_i's and S. Unfortunately finding S is not easy, searching for a convenient big prime is already a problem and factoring in $\mathbb{F}_p[X]$ with p large can be expensive.

To overcome this difficulty, we can try to take a small prime p and an integer l so that p^l is large enough. To simplify, let us imagine that we have a decomposition $f \equiv gh \bmod p$. We want to obtain a factorization $f \equiv f^* g^* \bmod p^l$. It is easy to see that we are done if we can answer to the following question:

Let R be a commutative ring with unity 1, $m \in R$ and $f \in R[X]$, $g, h \in R[X]$ coprime such that $f \equiv gh \bmod m$. Can we "lift" this to a coprime factorization $f = g^* h^* \bmod m^2$?

Imagine that $sg + th = 1$ is a Bezout relation between g and h. Then if we put $e = f - gh$, $g^* = g + te$ and $h^* = h + se$ we have trivially our conclusion. The problem now comes from the degrees of our new polynomials.

Example 3.16. Take $x^4 - 1 = (x-2)(x^3 + 2x^2 - x - 2) \bmod 5$, $s = -2$ and $t = 2x^2 - 2x - 1$. We obtain $g^* = 10x^4 - 9x^3 - 13x^2 + 9x + 3$ and $h^* = -10x^2 + x + 8$. We have of course $f \equiv g^* h^* \bmod 25$ but the degrees of f^* and g^* are too large.

To overcome this problem we use division with remainder in $R[X]$ which is possible if we divide by *monic* polynomials. If we use this trick we obtain the following algorithm in which $G = et \pmod{m^2, g}$ means that we consider the remainder of the division of et by g modulo m^2. Let us note that the algorithm must return a new Bezout relation if we want to iterate it.

Algorithm 10. Hensel step

```
Input: R (commutative, unital), $f, g, h \in R[X]$ monic, such that $\deg s < \deg h$, $\deg t < \deg g$, $f \equiv gh \pmod{m}$, $sg + th \equiv 1 \pmod{m}$.
Output: $g^*, h^* \in R[X]$ monic, $s^*, t^* \in R[X]$ such that $\deg s^* < \deg h^* = \deg h$, $\deg t^* < \deg g^* = \deg g$, $f \equiv g^* h^* \pmod{m^2}$, $s^* g^* + t^* h^* \equiv 1 \pmod{m^2}$.
1: Let $e = f - gh$,
       $G = et \pmod{m^2, g}$,     $g^* = g + G$
       $H = es \pmod{m^2, h}$,     $h^* = h + H$
2: Let $e = 1 - (sg^* + th^*)$
       $S = et \pmod{m^2, h^*}$,   $s^* = s + S$
       $T = es \pmod{m^2, g^*}$,   $t^* = t + T$
```

Proof. Letting $g^* = g + G$, $h^* = h + H$, $s^* = s + S$, $t^* = t + T$, we look for $G, H, S, T \equiv 0 \pmod{m}$ with $\text{deg}G < \text{deg}g$, $\text{deg}H < \text{deg}h$, $\text{deg}S < \text{deg}s$, $\text{deg}T < \text{deg}t$.

First lift g and h. We want

$$g^*h^* \equiv gh + Hg + Gh \pmod{m^2} \equiv f.$$

Letting $e = f - gh \equiv 0 \pmod{m}$, we try to solve $Hg + Gh \equiv e \pmod{m^2}$.

The quotient ring R/m^2 is not a field so we cannot solve this in the usual way (extended Euclidean algorithm). Fortunately, a Bezout relation is provided with the input, and $G = et$, $H = es$ is a solution. Unfortunately, we want $\deg G < \deg g$ and $\deg H < \deg h$.

But we can modify H by a multiple of h and G by the corresponding multiple of g, so write: $G = qg + r$, $\deg r < \deg g$ (division possible because g is monic), and replace $G \leftarrow G - qg$, $H \leftarrow H + qh$.

1. $Hg + Gh$ does not change.

2. $G = r$ satisfies $\deg G < \deg g$.

3. $\deg e < \deg f = \deg g + \deg h$. (Since f, g, h are monic, leading terms cancel). Now use $Gh + Hg = e$; since $\deg Gh = \deg G + \deg h < \deg g + \deg h$, we finally obtain $\deg(Hg) < \deg g + \deg h$. In other words, $\deg H < \deg h$ as required.

4. $f \equiv g^*h^* \pmod{m^2}$.

Now lift s, t: we have

$$s^*g^* + t^*h^* = sg^* + th^* + Sg^* + Th^*, \quad sg^* + th^* \equiv 1 \pmod{m},$$

and we solve $Sg^* + Th^* \equiv 1 - (sg^* + th^*) \pmod{m^2}$ exactly as before. □

Example 3.17. With the polynomials of example 3.16 we obtain $g^* = x^3 + 7x^2 - x - 7$, $h^* = x - 7$, $s^* = 8$ and $t^* = -8x^2 - 12x - 1$.

Corollary 3.18 (Hensel's lemma)**.** *Given a nonzero $p \in R$ and an integer $\ell \geq 1$, and assuming the input specification of algorithm* 10 *for $m = p$, we can compute polynomials as in the output specification, but with m^2 replaced by p^ℓ.*

Proof. Apply Hensel step inductively for $m = p, p^2, p^4, \ldots$. □

Remark 3.19. The most important applications are $R = \mathbb{Z}$ and p prime of course, but also $R = K[Y]$ and $p = X$. In general Hensel's lemma is known as the result in the specific first case.

Lemma 3.20. *If $R = \mathbb{Z}$, $\deg f = n$, and all inputs satisfy $\| \cdot \|_\infty < m^2$, the cost of a Hensel step is $\widetilde{O}(n \log m)$, i.e. essentially linear in the input size.*

Proof. For any arithmetic operation of polynomials in $\mathbb{Z}[X]$ occuring in the algorithm (whose degrees are at most $2n$ and coefficients less than m^4) the cost is in $O(M(n))$ operations in $\mathbb{Z}$ on numbers of length in $O(\log m)$. Such an arithmetic operation in $\mathbb{Z}$ has a word complexity in $O(M(\log m))$ and taking the usual $M(k) = \widetilde{O}(k)$ gives the result. □

Algorithm 11. Hensel multi-lift

Input: R commutative ring, $p \in R$ such that R/p is a field, $\ell \in \mathbb{Z}_{>0}$. $f \in R[X]$ monic. $f_1, f_2, \ldots, f_r$ monic in $R[X]$, pairwise coprime in $(R/p)[X]$ such that

$$f = \prod_{i=1}^{r} f_i \pmod p.$$

Output: Monic $f_1^*, \ldots, f_r^* \in R[X]$ such that

$$f \equiv \prod_{i=1}^{r} f_i^* \pmod{p^\ell}.$$

```
1: If $r = 1$, return $f$.
2: Let $k = \lfloor r/2 \rfloor$, $g = f_1 \ldots f_k$, $h = f_{k+1} \ldots f_r$, and find $s, t$ such that
   $sg + th \equiv 1 \pmod p$ (Extended Euclidean Algorithm).
3: Compute $g^*, h^*$ such that $f \equiv g^* h^* \pmod{p^\ell}$, using $O(\log \ell)$ Hensel Steps
   $p \to p^2$, $p^2 \to p^4$, etc..
4: Call ourselves recursively with $g^*, (f_1 \ldots f_k)$, then $h^*, (f_{k+1} \ldots f_r)$ and
   return the concatenation of the factors.
```

Example 3.21. Take $f = x^4 - 1$ again. We have

$$x^4 - 1 \equiv (x-1)(x-2)(x+1)(x+2) \bmod 5,$$

and applying the algorithm with $\ell = 4$ we obtain

$$x^4 - 1 \equiv (x-1)(x-182)(x+182)(x+1) \bmod 625.$$

Theorem 3.22. *Assume $R = \mathbb{Z}$, $\deg f = n$, and all inputs are reduced modulo p^ℓ. Then the lifting cost is $\widetilde{O}(n \log p^\ell)$, essentially linear.*

Proof. See Gerhard and Von Zur Gathen. □

3.2.4 Infinite version

There is also an infinite version of Hensel's result which can be formulated in p-adic terms.

$\mathbb{R}$ is a convenient "limit field" associated to convergent sequences of rational approximations: for all $x \in \mathbb{R}$, there exists $\widehat{x}(\varepsilon) \in \mathbb{Q}$ such that $|x - \widehat{x}| < \varepsilon$. From a computational point of view we only know (some of) the approximations $\widehat{x}$.

We need an analogous ring associated to a different metric: now we want closer and closer approximations modulo ever larger powers of a given prime p: for all $x \in \mathbb{Z}_p$ there exists $\widehat{x}(n) \in \mathbb{Z}$ such that $|x - \widehat{x}|_p < p^{-n}$.

Definition 3.23. Let $|\cdot|_p : \mathbb{Q} \to \mathbb{R}^+$ defined by $x \mapsto p^{-v_p(x)}$ for $x \neq 0$ and $|0|_p = 0$. It is a non-Archimedean absolute value, satisfying $|x + y|_p \leq \max(|x|_p, |y|_p)$.

Definition 3.24. Equip $\mathbb{Q}$ with the topology afforded by the $|\cdot|_p$ metric. Let $\mathbb{Q}_p$ be the ring of Cauchy sequences in $\mathbb{Q}^{\mathbb{N}}$, modulo the ideal of sequences converging to 0. (Both "Cauchy sequence" and "converging to 0" are understood with repect to the p-adic metric!) The absolute value $|\cdot|_p$ extends to $\mathbb{Q}_p$ (with values in $\mathbb{Q}$), making it a topological *field*.

Definition 3.25. Let $\mathbb{Z}_p = \{x \in \mathbb{Q}_p;\ |x|_p \leq 1\}$ be the unit ball in $\mathbb{Q}_p$. This is a compact ring, whose field of fraction is $\mathbb{Q}_p$, in fact $\mathbb{Q}_p = \mathbb{Z}_p[1/p]$. $\mathbb{Z}_p$ is a principal local ring with unique maximal ideal $p\mathbb{Z}_p = \{x \in \mathbb{Q}_p;\ |x|_p < 1\}$ the open unit ball. One proves that the natural map $\mathbb{Z}_p/p^k\mathbb{Z}_p \to \mathbb{Z}/p^k\mathbb{Z}$ is an isomorphism.

An alternative construction for $\mathbb{Z}_p$ and $\mathbb{Q}_p$ is

$$\mathbb{Z}_p = \varprojlim_{n \geq 1} \mathbb{Z}/p^n\mathbb{Z},$$

that is the subring of $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p^2\mathbb{Z} \times \cdots$, containing the $(x_1, x_2, \dots)$ such that $x_i \equiv x_j \pmod{p^i}$, for all $i \leq j$. Then define $\mathbb{Q}_p = \mathrm{Frac}\mathbb{Z}_p = \mathbb{Z}_p[1/p]$.

The important property for us is that $\mathbb{Z}$ is dense in $\mathbb{Z}_p$ (and $\mathbb{Q}$ in $\mathbb{Q}_p$). We represent an $x \in \mathbb{Z}_p$ by an approximation $\widehat{x} \in \mathbb{Z}$ such that $x \equiv \widehat{x} \pmod{p^\ell}$ and say that x is known modulo p^ℓ. The next theorem is the crux of all modern factorization algorithms over $\mathbb{Q}[X]$: as already seen, one can refine an approximate factorization in *coprime* factors to an arbitrary accuracy, and this can be reformulated in the more elegant following way.

Theorem 3.26 (Hensel's lemma – infinite version)**.** *Let $T \in \mathbb{Z}_p[X]$ be a monic polynomial, and a collection of polynomials $T_i \in \mathbb{Z}_p[X]$, $i \leq r$, which are monic and pairwise coprime, such that*

$$T \equiv \prod_{i=1}^{r} T_i \pmod{p} \quad (= \bmod p\mathbb{Z}_p[X]).$$

(This is an equality in $\mathbb{Z}_p[X]/p\mathbb{Z}_p[X] \simeq \mathbb{F}_p[X]$.) There exist unique $T_i^* \in \mathbb{Z}_p[X]$ such that $T = \prod_{i=1}^{r} T_i^*$ and $T_i^* \equiv T_i \pmod{p}$ for $i \leq r$.

Proof. Let $k > 0$. Starting from a factorization $T = \prod_{i=1}^{r} T_i \pmod{p^k}$, we construct explicitly $t_i \in \mathbb{Z}_p[X]$, $\deg t_i < \deg T_i$ such that

$$T = \prod_{i=1}^{r} (T_i + t_i p^k) \pmod{p^{2k}}.$$

In fact, from a Bezout relation in the principal ring $\mathbb{Q}_p[X]$, we obtain $t_i \in \mathbb{Q}_p[X]$, $\deg t_i < \deg T_i$ such that

$$\sum_{i=1}^{r} t_i \prod_{j \neq i} T_j = \frac{T - \prod_{i=1}^{r} T_i}{p^k} \in \mathbb{Z}_p[X].$$

We claim the t_i belong to $\mathbb{Z}_p[X]$. If not, let t_i have the largest denominator p^d; clear denominators by multiplying by p^d and reduce modulo p: we see that T_i divides $p^d t_i$ modulo p, hence $t_i = 0$, a contradiction.

By induction this yields $T_i^{(k)} \in \mathbb{Z}_p[X]$, $\deg T_i^{(k)} \leq \deg T$ such that $T_i^{(k)} \equiv T_i^{(\ell)} \pmod{\ell}$ for all $\ell \leq k$, i.e. $k \mapsto T_i^{(k)}$ is a Cauchy sequence in $\mathbb{Z}_p[X]_{\leq \deg T}$, which is complete. Therefore, it converges to $T_i^* \in \mathbb{Z}_p[X]$. To prove unicity, notice that $\mathbb{Z}_p$ is a UFD (it is principal!), hence $\mathbb{Z}_p[X]$ is also a UFD. □

3.2.5 Zassenhaus' algorithm

As seen above, our goal is to factor a monic squarefree polynomial $T \in \mathbb{Z}[X]$ over $\mathbb{Z}$.

Definition 3.27. Le p prime, $k \in \mathbb{Z}_{>0}$ and R one of the rings $\mathbb{Z}$, $\mathbb{Z}_p$ or $\mathbb{Z}/p^i\mathbb{Z}$ for some $i \geq k$. If $a \in R$, we write $\mathrm{MOD}(a, p^k)$ for the unique representative in $\mathbb{Z}$ of $a \pmod{p^k}$, which lies in $]-p^k/2, p^k/2]$. We extend this definition coefficient-wise to polynomials in $R[X]$.

Algorithm 12. Zassenhaus' algorithm

```
Input: $T \in \mathbb{Z}[X]$, monic, squarefree
Output: The monic irreducible factors of $T$ in $\mathbb{Q}[X]$.
Pick $p$ prime such that $T \in \mathbb{F}_p[X]$ remains squarefree and compute
    $$T \equiv \prod_{i=1}^{r} T_i \pmod{p}$$
  where the $T_i$ are distinct, monic and irreducible in $\mathbb{F}_p[X]$.
Let $B = 2^{\det T} \| T \|_2$   {= bound for the sup-norm of a factor in $\mathbb{Z}[X]$}
Compute $\ell$ minimal such that $p^\ell > 2B$ and lift
    $$T \equiv \prod_{i=1}^{r} T_i^* \pmod{p^\ell}.$$
Let $S_0 = \{1, \dots, r\}$.
for $S \subset S_0$, by increasing size do
    Compute
        $$A_S = \mathrm{MOD}\Big(\prod_{i \in S} T_i^*, p^\ell\Big) \in \mathbb{Z}[X].$$
    if $A_S \mid T$ then
        Output $A_S$.  {$A_S$ is an irreducible factor}
        Replace $S_0 \leftarrow S_0 \setminus S$ and $T \leftarrow T/A_S$.
```

Proof. We first prove that all factors appear as A_S for some S: by Corollary 3.15 a factor Q satisfies $\| Q \|_\infty \leq B$, and we certainly have $Q \equiv A_S \pmod{p^\ell}$ for a unique S (since the T_i are distinct: apply Hensel Lemma and reduce mod p^ℓ the factorization in $\mathbb{Z}_p[X]$). Hence

$$\| Q - A_S \|_\infty \leq B + \frac{p^\ell}{2} < p^\ell,$$

by the choice of ℓ. Since this polynomial is divisible by p^ℓ, it is identically 0 and $Q = A_S$.

It remains to prove that the A_S we output are irreducible. This is guaranteed by the order in which we consider the set S: if A_S is not irreducible, then $A_S = BC$ in $\mathbb{Q}[X]$, for non-constant B and C. By what we have just proved B is of the form $A_{S'}$ for a proper subset S' of S, which cannot occur because smaller sets are considered first and the indices in S' would already have been removed from S_0. □

Remark 3.28. In the loop over the subsets S, we may stop as soon as $\sharp S > \sharp S_0/2$, since A_S is a factor if and only if $A_{S'}$ is a factor, with S' the complement of S in S_0. In particular, we consider at most 2^{r-1} sets S.

Remark 3.29. As already said, there is a "big prime" variant of this algorithm where we directly chose $p > 2B$ and skip the Hensel lift (take $\ell = 1$). But this is a bad idea since Hensel lifting is essentially linear time ($\widetilde{O}(n \log p^\ell)$) whereas factorization in $\mathbb{F}_p[X]$ for big p definitely is not: $\widetilde{O}(n^2 \log p + n \log^2 p)$ using the fastest randomized algorithms from Section 3.1.6. Also it is not obvious a priori how to construct a "big prime", whereas small primes are easy to find using a sieve. (See later.)

We now examine the various costs involved, letting $n = \deg T$, $A = \|T\|_\infty$. We make use of the following classical result in analytic number theory (see Tenenbaum)

Theorem 3.30 (Prime Number Theorem (PNT)). *As $x \to +\infty$, we have*

$$\Theta(x) := \sum_{p \le x} \log p \sim x.$$

Effectively, one has $\Theta(x) > 0.98x$ for $x > 7481$.

The usual, more transparent, formulation is in terms of the function $\pi(x) := \sharp\{p \le x\}$, satisfying $\pi(x) \sim x/\log x$. The above function is the one which turns up in our case. Using integration by part (Abel's summation), it is not dificult to show that the two forms of the PNT are equivalent.

Theorem 3.31. *There exists a prime $p = O(n \log(nA))$ such that T is squarefree in $\mathbb{F}_p[X]$. In particular $\log p = O(\log n + \log\log A)$.*

Proof. Let $D = \operatorname{Res}(T, T')$. This is an integer (since $T, T' \in \mathbb{Z}[X]$), which is non-zero since T is squarefree in $\mathbb{Q}[X]$, hence $\gcd(T, T') = 1$. Since D is a polynomial in the coefficients of T and T' and reduction mod p is a ring homomorphism, the resultant in $\mathbb{F}_p[X]$ of $\bar{T}$ and $\bar{T}'$ is $\bar{D}$, and this is non-zero if and only if $\bar{T}$ is squarefree in $\mathbb{F}_p[X]$.

By Hadamard's bound and $\deg T' \le n - 1$, $|D| \le \|T\|_2^{n-1} \|T'\|_2^n$, from which we obtain $\log |D| = O(n \log(nA))$. Assume now that all primes $p \le x$ divide D, for some $x > 7481$. Then

$$\log |D| \ge \sum_{p \le x} \log p > 0.98x,$$

which is a contradiction for $x \gg n \log(nA)$. □

Corollary 3.32. *The cost of finding a suitable p is $\widetilde{O}(n^2 \log^2 A)$*

Proof. Reducing the $n+1$ coefficients of T modulo p costs $O(n \log A \log p)$. Once the inputs have sup norm less than p, computing $\gcd(\bar{T}, \bar{T}')$ costs $\widetilde{O}(n \log p)$, negligible before the reduction cost. Summing this for all $p = \widetilde{O}(n \log A)$, and using the PNT, we obtain $\widetilde{O}(n^2 \log^2 A)$. □

In practice, one chooses a few random primes less than *twice* an explicit upper bound for $|D|$ (T and T' are known!). This guarantees that less than half the primes up to the bound are unsuitable, so we expect ≤ 2 trials before hitting one, replacing the above estimate by $\widetilde{O}(n \log A)$ (no longer deterministic). Even after hitting a suitable prime, we still compute a few more and choose the one such that the number of modular factors r is minimal.

Another advantage is that it can give informations on the number and the degrees of the factors and sometimes prove directly that the polynomial is irreducible (see exercise).

Note that p is so small that we can factor deterministically in $\mathbb{F}_p[X]$ (in time exponential in $\log p$) and still remain polynomial-time with respect to n and $\log A$. This is important for our later theoretical result that factoring in $\mathbb{Q}[X]$ can be done in deterministic polynomial time. In practice, one uses a fast randomized algorithm, for a cost of $\widetilde{O}(n^2 \log p + n \log^2 p)$:

Corollary 3.33. *The (randomized) cost of factoring over $\mathbb{F}_p$ is $\widetilde{O}(n^2 \log\log A + n(\log\log A)^2)$.*

Corollary 3.34. *The cost of the Hensel lift is $\widetilde{O}(n\ell \log p) = \widetilde{O}(n^2 + n \log A)$.*

Proof. We have $B \leq 2^n \sqrt{n+1} A$ and want $p^\ell > 2B$, i.e. $\ell = \log(2B)/\log p + O(1)$. Hensel lifting costs

$$\widetilde{O}(n\ell \log p) = \widetilde{O}(n \log(2B) + n \log p).$$

Using the upper bound $\log B = O(n + \log(A))$ and $\log p = O(\log n + \log\log A)$, we obtain $\widetilde{O}(n^2 + n \log A)$. □

So far, so good. Unfortunately, the number of sets S to consider is a priori exponential in n. In fact, if T is irreducible, we have to consider 2^{r-1} sets. And for *bad* polynomials, r can be as large as $n/2$, independently of p. Take T the minimal polynomial of

$$\alpha_k = \sqrt{2} + \sqrt{3} + \cdots + \sqrt{p^k},$$

where p_k is the k-th prime number. The polynomial T is irreducible of degree $n = 2^k$, and $\mathbb{Q}(\alpha_k)/\mathbb{Q}$ is Galois with group $G = (\mathbb{Z}/2\mathbb{Z})^k$. A theorem of Frobenius relates the cycle structure of elements of G (viewed as conjugacy classes in S_n) with the factorization of T modulo p. In this example, it asserts that T has either n or $n/2$ factors in $\mathbb{F}_p[X]$, whenever it is squarefree.

More generally, if T is irreducible, generating a Galois extension of $\mathbb{Q}$ with Galois group of exponent e, we have $r \geq n/e$. Hence Zassenhaus algorithm runs in deterministic polynomial time up to the last loop, which unfortunately is exponential time in n. (But not in $\log A$: for small degrees, it is fine.)

On the other hand, *if* there exist primes modulo which r (number of modular factors) is not much bigger than s (number of true factors), then effective forms of the theorem of Chebotarëv tell us that a relatively small such p exist.

3.2.6 LLL improvement

This provides a replacement for the last step in Zassenhaus algorithm, avoiding the exhaustive enumeration which make it exponential time in the worst case. This results in a deterministic polynomial time algorithm, though not a very practical one.

Lemma 3.35. *Let $f, g \in \mathbb{Z}[X]$ have positive degrees. Suppose $u \in \mathbb{Z}[X]$ is non-constant, monic, and divides f and g in $(\mathbb{Z}/m\mathbb{Z})[X]$, for some $m > |\mathrm{Res}(f, g)|$. Then $\gcd(f, g) \in \mathbb{Q}[X]$ is non-constant.*

Proof. If $\gcd(f, g)$ is constant, there exist $s, t \in \mathbb{Z}[X]$ such that $sf + tg = \mathrm{Res}(f, g) = R \in \mathbb{Z}$ (use Cramer's formulas see why we can take s and t in $\mathbb{Z}[X]$). Hence $\bar{u}$ divides $\bar{R}$ in $(\mathbb{Z}/m\mathbb{Z})[X]$; since u is monic, non-constant, $\bar{u}$ has degree ≥ 1 which implies $\bar{R} = 0$. Since $m > |R|$, $R = 0$ and f, g have a common factor in $\mathbb{Q}[X]$. □

The idea is to use the lemma together with the LLL algorithm as follows: let $m = p^\ell$, $f = T \in \mathbb{Z}[X]$ of degree n and $u \in \mathbb{Z}[X]$ which is an irreducible factor of T modulo m, e.g. $u = T_1^*$ from Zassenhaus algorithm. Then look for a $g \in \mathbb{Z}[X]$ of degree $k < n$ such that

- $u \mid g$ modulo m,
- $\|g\|_2^n < m \, \|T\|_2^{-k}$, which implies $|\mathrm{Res}(T, g)| < m$ by Hadamard.

If T is reducible, then there exists such a "small" g, provided m is large enough. How large ? We know $k \leq n - 1$ and $\|g\|_2 \leq B$, hence want $m > \|T\|_2^{n-1} B^n$. We only need to express the divisibility condition with a lattice and we have a short vector problem! From any short vector, the lemma finds a factor. On the other hand if no short vector is found, or the short vector does not yield a factor, it proves that T was irreducible.

So let $j \geq \deg u$ and $\Lambda_j \subset \mathbb{Z}[X]$ be the lattice (of rank j) generated by the

$$\left\{uX^i, i < j - \deg u\right\} \cup \left\{mX^i, i < \deg u\right\}$$

Lemma 3.36. *$g \in \Lambda_j$ if and only if $\deg g < j$ and u divides g modulo m.*

Lemma 3.37. *Let $m \geq 2^{n^2/2} B^n$, and g the first vector in an LLL-reduced basis of Λ_{n-1}. Then either $\gcd(g, T)$ is a non trivial factor of T or T is irreducible.*

Proof. Assume T reducible. Then there exists h in Λ_j which divides T in $\mathbb{Z}[X]$ (take the irreducible rational factor containing T_1^*). From Landau's bound, we know $\|h\|_2 \leq B$. Let $j = n - 1$; by the properties of LLL-reduced bases,

$$\|g\|_2 \leq 2^{(j-1)/2} \|h\|_2 \leq 2^{n/2} B.$$

Then Lemma 3.35 ensures we get a factor provided

$$\|g\|_2^n \|T\|_2^{n-1} < m.$$

Now use the trivial bound $\|T\|_2 \leq B$. □

Corollary 3.38. *In order to apply the LLL method we must lift modulo p^ℓ, where $\log p^\ell = O(n^2 + n \log A)$, for a cost $\widetilde{O}(n(n^2 + n \log A))$.*

Note that this lifting bound is n times larger than Landau's bound used in Zassenhaus algorithm.

Theorem 3.39. *Let $T \in \mathbb{Z}[X]$ be monic. We can factor T in $\mathbb{Q}[X]$ in deterministic polynomial time $\widetilde{O}(n^{10} + n^8 \log^2 A)$.*

Proof. An LLL reduction costs $\widetilde{O}(n^5 \log^2 m) = \widetilde{O}(n^9 + n^7 \log^2 A)$, the final modular gcd has negligible cost $\widetilde{O}(n^2 + n \log B)$. We must perform the above at most $2n$ times since T has at most n factors (so at most n splittings and at most n failures denoting an irreducible polynomial). □

A practical improvement is to take for u some T_i^*, then consider successively $\Lambda_{\deg u}$, $\Lambda_{2 \deg u}$, …, $\Lambda_{2^k \deg u}$, trying to guess the degree of a factor g containing u by dichotomy. The cost now becomes $\widetilde{O}(d^9 + d^7 \log^2 A)$ to find a factor of degree d. Such a factor may be reducible, but any factor containing u has degree $> d/2$. By dichotomy (knowing a factor of degree d_2 and that no factor of degree d_1 exist, we try for degree $(d_1 + d_2)/2$) we arrive at the irreducible factor of degree d containing u in $O(\log d)$ steps, and the cost remains $\widetilde{O}(d^9 + d^7 \log^2 A)$. Since $\sum d = n$, we have $\sum d^t \le (\sum d)^t = n^t$ for any $t > 0$ and the total cost goes down to $\widetilde{O}(n^9 + n^7 \log^2 A)$.

Another improvement is to try smaller lattices or lift to smaller accuracy than p^ℓ: if the polynomial is reducible, we may get lucky and find a factor. Unfortunately, when the polynomial is irreducible, there is no way to avoid computing up to the worst case bounds. This is still horrendously expensive for large degrees, e.g. $n = 1000$: 10^{27} operations are beyond the capabilities of any current computer.

3.2.7 Van Hoeij's algorithm

Van Hoeij's algorithm (2002), in the version we shall describe, has the same *proven* complexity as LLL (in particular polynomial time) but is orders of magnitude faster in practice. Moreover, it gives all the factors at once, whereas LLL gives just one. Variants of van Hoeij's method can be proven to perform better than LLL, but they are more technical to prove and describe so we will be content with heuristics at this point. The setup is the same as for Zassenhaus and LLL:

Input: $f \in \mathbb{Z}[X]$, monic, square-free, of degree n, $f = \prod_{i=1}^r g_i \in \mathbb{Z}_p[X]$. The $g_i \in \mathbb{Z}_p[X]$ are monic, irreducible, pairwise distinct, and MOD $\big(g_i, p^\ell\big) \in \mathbb{Z}[X]$ is known for all i.

Output: $f = \prod_{i=1}^s f_j \in \mathbb{Q}[X]$, $s \le r$. The $f_j \in \mathbb{Z}[X]$, are monic and irreducible.

Notations

Let $G_p = \langle g_1, \cdots, g_r \rangle$ be the multiplicative subgroup of $\mathbb{Q}_p(X)^*$ generated by the g_i and $G_\mathbb{Q} = \langle f_1, \cdots, f_s \rangle$ the subgroup generated by the f_j. Note that $G_\mathbb{Q}$ is a subgroup of G_p and they both are free $\mathbb{Z}$-modules of rank r and s respectively. Our goal is to compute $G_\mathbb{Q}$; this is sufficient because the HNF algorithm enables us to deduce the f_j from any $\mathbb{Z}$-basis of $G_\mathbb{Q}$:

Lemma 3.40. *Using (g_i) as a fixed basis for G_p, a suitable permutation of the (f_j) form an HNF basis for $G_\mathbb{Q}$. More precisely, if $f_j = \prod_i g_i^{h_{i,j}}$ and $H = (h_{i,j})$ is a matrix with $\{0,1\}$-coefficients then, up to a reordering of its columns, H is in HNF.*

Proof. This is true for any matrix with $\{0,1\}$ coefficients containing a most a single 1 on each line. □

Corollary 3.41. *If the $b_j = \prod_i g_i^{a_{i,j}}$, $j \le s$, form a basis of $G_{\mathbb{Q}}$, then in the notations of the Lemma, H is the HNF of $(a_{i,j})$.*

The idea is to find a basis of $G_{\mathbb{Q}}$ as a set of short vectors. First we linearize the problem. Let ϕ be the following map:

$$\begin{array}{rcl} \phi\colon (G_p, \times) & \to & (\mathbb{Q}_p(X), +) \\ g & \mapsto & f\frac{g'}{g} \end{array}$$

From the derivative laws $(fg)' = f'g + g'f$ so $\frac{(fg)'}{fg} = \frac{f'}{f} + \frac{g'}{g}$ and ϕ is a morphism. In fact,

$$\phi\Big(\prod_{i=1}^{r} g_i^{e_i}\Big) = \sum_{i=1}^{r} e_i g_i' \frac{f}{g_i} \in \mathbb{Q}_p[X],$$

with degree $< n$. Analogously, we see that $\phi\left(G_{\mathbb{Q}}\right) \subset \mathbb{Z}[X]_{<n}$.

More generally, define

$$\Phi(g_i) = \phi(g_i) + X^{n+r-i},$$

and extend it additively to G_p. In effect, we concatenate an identity matrix above the matrix giving the $\phi(g_i)$. This will keep track of column operations; in a short vector algorithm, it further ensures that perturbations of the input remain small.

It is difficult to work with $\phi\left(G_p\right)$ because it is not a $\mathbb{Z}$-module of finite type, and the g_i are not known exactly. So we introduce the lattice Λ, which is the $\mathbb{Z}$-module generated by $\Phi(g_i)$, $i \le r$ and by $p^\ell X^i$, $i \le n$. Expressed on the natural basis $1, X, \ldots, X^{n+r-1}$, Λ is generated by the columns of the following matrix:

$$\begin{pmatrix} \mathrm{Id}_r & 0 \\ A & p^\ell \mathrm{Id}_n \end{pmatrix},$$

where A is the $n \times r$ matrix of the MOD $\left(\phi(g_i), p^\ell\right)$, written by decreasing degree. (Recall that $\phi(g_i)$ has degree $< n$.) A generic vector in Λ has the form

$$\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_r \\ \sum_i e_i \mathrm{MOD}\left(\phi(g_i), p^\ell\right) + p^\ell Q \end{pmatrix}$$

with $Q \in \mathbb{Z}[X]_{<n}$. The above vector is $\Phi(\prod g_i^{e_i})$, where the part of degree $< n$ is taken modulo p^ℓ, and

$$\left(\Phi(G_p) + p^\ell \mathbb{Z}_p[X]_{<n}\right) \cap \mathbb{Z}[X] = \Lambda.$$

The $\Phi(f_j)$ have $(e_1, \ldots, e_r)$ in $\{0, 1\}$. Compared to LLL, the base change matrix is very simple: in LLL it is given by the coefficients of a factor, which may be large. We shall prove that, if the accuracy p^ℓ is large enough, the first s vectors in an LLL-reduced basis of Λ will form a basis of $\Phi(G_{\mathbb{Q}})$, enabling us to find $G_{\mathbb{Q}}$ and the rational factors. The recipe to find s is simple: eliminate the big basis vectors. But first we need some lemmas:

Lemma 3.42 (Mahler, 1961)**.** *Let $A \in \mathbb{C}[X]$, then $M(A') \leq \deg(A)M(A)$.*

Proof. The proof is surprisingly difficult and we shall only sketch it, see [10, Appendix D] for details. Letting $\alpha_1, \ldots, \alpha_d$ and $\beta_1, \ldots, \beta_{d-1}$ be the complex roots of A and A' respectively, we must prove

$$\prod_{j=1}^{d-1} \max(1, |\beta_j|) \leq \prod_{j=1}^{d} \max(1, |\alpha_j|).$$

For given $\alpha_1, \ldots, \alpha_{d-1} \in \mathbb{C}$, and $t \in \mathbb{R}$, let

$$f_t(\alpha_d) = \sum_{j \leq d} \frac{1}{e^{2i\pi t} - \alpha_j}.$$

Using Jensen's formula, Mahler's measure admits the alternative expression

$$\log M(A) = \int_0^1 \log \left| A(e^{2i\pi t}) \right| \, dt.$$

(The integral is well-defined, since any singularity is of the form log(polynomial).) Using

$$\frac{A'}{A}(z) = \sum_{j \leq d} \frac{1}{z - \alpha_j},$$

a simple modification of the proof shows that

$$\int_0^1 \log |f_t(\alpha_d)| \, dt = \log \frac{M(A')}{M(A)} = \log d + \log \prod_{j=1}^{d-1} \max(1, |\beta_j|) - \log \prod_{j=1}^{d} \max(1, |\alpha_j|).$$

Hence the inequality to prove becomes equivalent to

$$\int_0^1 \log |f_t(\alpha_d)| \, dz \leq \log d. \tag{3.3}$$

The function $\alpha_d \mapsto \log |f_t(\alpha_d)|$ is subharmonic on $\mathbb{C} \setminus \{z \colon |z| = 1\}$, hence so is

$$\alpha_d \mapsto \int_0^1 \log |f_t(\alpha_d)| \, dt.$$

From the maximum modulus principle for subharmonic functions applied to $\{z\colon |z| < 1\}$ and $\{z\colon |z| > 1\}$, it follows that the integral in (3.3) above is maximal for some α_d with $|\alpha_d| = 1$. By symmetry, it is enough to prove the inequality when $|\alpha_1| = \cdots = |\alpha_d| = 1$.

In that case, it follows immediately from Lucas's theorem: the zeroes of A' are in the convex hull of the zeroes of A, which implies that $|\beta_j| \leq 1$ for all j. To prove Lucas's result, let β be a root of A', then

$$0 = \frac{A'}{A}(\beta) = \sum_{j=1}^{d} \frac{1}{\beta - \alpha_j} = \sum_{j=1}^{d} \frac{\overline{\beta - \alpha_j}}{|\beta - \alpha_j|^2}.$$

Taking the conjugate, it follows that β is a linear combination of the α_j with *positive* coefficients. □

Lemma 3.43. *If $g \in \mathbb{Z}[X]$ divides f, then $\|\phi(g)\|_2 \leq n2^n \cdot \|f\|_2$*

Proof. As in Landau's theorem, let $M(A)$ be the Mahler measure of A. For two polynomials A, B, we have $M(AB) = M(A)M(B)$. Remember that $M(A) \leq \|A\|_2 \leq 2^{\deg A} M(A)$. Since $\deg \phi(g) < n$ and $\frac{f}{g}$ is a polynomial, we have

$$\|\phi(g)\|_2 \leq 2^n \cdot M(fg'/g) = 2^n M(g')M(f/g).$$

Using Mahler's lemma and Landau's estimate,

$$M(g')M(f/g) \leq n2^n \cdot M(g)M(f/g) = n2^n M(f) \leq n2^n \cdot \|f\|_2 \,.$$

□

Corollary 3.44. *For all $j \leq s$, $\|\Phi(f_j)\|_2 \leq n2^n \|f\|_2 + \sqrt{r}$.*

Proof. Exercise. □

For $g = \prod_i g_i^{e_i} \in G_p$, we define the *support* of g by

$$\mathrm{Supp}(g) = \{i\colon e_i \neq 0\}\,.$$

Lemma 3.45. *$g_i \mid \phi(g) \Leftrightarrow i \notin \mathrm{Supp}(g)$.*

Proof. Since ϕ is a morphism, we have $\phi(g) = \sum_i e_i \phi(g_i)$. Now, $g_i \mid \phi(g_j)$ if and only if

$$g_i \mid \frac{g_j'}{g_j} \cdot f = \prod_{k \neq j} g_k \cdot g_j'$$

Since g_j is irreducible, and $\mathbb{Q}_p$ has characteristic 0, g_j' and g_j are coprime. Hence $g_i \mid \phi(g_j) \Leftrightarrow i \neq j$. Since g_i divides all the summands in $\phi(g)$ except the i-th one, $g_i \mid \phi(g)$ if and only if the latter is 0, i.e. $e_i = 0$. □

Lemma 3.46. *Let G be a subgroup of G_p containing strictly $G_{\mathbb{Q}}$ and $u \in G \setminus G_{\mathbb{Q}}$. Then there exists $g \in G \setminus G_{\mathbb{Q}}$ such that:*

1. *$g_i \mid \phi(g)$ for some $i \leq r$*
2. *$f_j \nmid \phi(g)$ for all $j \leq s$*
3. *$\operatorname{MOD}\left(\Phi(g), p^\ell\right)$ is not much larger than $\operatorname{MOD}\left(\Phi(u), p^\ell\right)$.*

Proof. We start with $g = u$ and modify it until the conditions hold. Because of Lemma 3.45, the first two conditions mean that $i \notin \operatorname{Supp} g$ and $\operatorname{Supp} f_j \cap \operatorname{Supp} g \neq \emptyset$ for all j.

So for all j such that $\operatorname{Supp}(f_j) \cap \operatorname{Supp}(g) = \emptyset$, we we replace g by gf_j. After this step, the second condition is satisfied. Since we have multiplied $u \notin G_{\mathbb{Q}}$ by elements in the subgroup $G_{\mathbb{Q}}$, the resulting g is not in $G_{\mathbb{Q}}$.

Write $g = \prod_i g_i^{e_i}$. We want to make sure that one of the e_i is zero. For $j \leq s$, let $S_j = \{e_i \colon i \in \operatorname{Supp} f_j\}$. There exists j such that $\#S_j > 1$, otherwise $S_j = \{s_j\}$ for all j and $g = \prod_j f_j^{s_j} \in G_{\mathbb{Q}}$, a contradiction.

Pick one $i \in S_j$, since g_i divides f_j to the first power, we replace g by $g/f_j^{e_i}$, ensuring that g_i does not divide g, without affecting the other conditions.

As for the last condition, the $\Phi(f_j)$ are bounded by Corollary 3.44, and so is e_i. □

Lemma 3.47. *Let (Λ, q) be a lattice, $(b_1, \ldots, b_n)$ a basis of Λ and $(b_1^*, \ldots, b_n^*)$ the associated Gram-Schmidt orthogonalized basis. Assume $q(b_j^*) > B$, for all $j > j_0$. Then if $v \in \Lambda$ satisfies $q(v) \leq B$, we have $v \in \langle b_1, \ldots, b_{j_0} \rangle_{\mathbb{Z}}$.*

Proof. Exercise (look at the proof of Theorem 2.37). □

This is true for any basis, but of course an LLL-basis is most suitable: since $2q(b_{i+1}^*) \geq q(b_i^*)$, the $q(b_i^*)$ do not decrease too fast. In fact, we expect them to increase quickly if the basis contains both small and large vectors. The idea is then that we expect non-rational inputs taken mod p^ℓ to yield big integers. (Just like we expect the average irrational number not to have a long sequence of 0 in its decimal development.)

Algorithm 13. van Hoeij's algorithm

```
1: Compute an LLL-reduced basis $(b_1, \ldots, b_m)$ of the lattice $\Lambda$ generated by
       $\left\{\operatorname{MOD}\left(\Phi(g_i), p^\ell\right), i \leq r\right\} \cup \left\{p^\ell X^i, i < n\right\}.$
2: Let $B = 2^n(n \|f\|_2 + \sqrt{r})$.
3: Let $s = j_0$ be minimal such that $\|b_j^*\|_2 > B$ for all $j > j_0$.   {we have
   $\Lambda_{\mathbb{Q}} \subset \langle b_1, \ldots, b_{j_0} \rangle_{\mathbb{Z}}$.}
4: Return $(b_1, \ldots, b_s)$
```

Theorem 3.48. *There exists an effective $C = O(n^2 + n \log \|f\|_2)$ such that if $\log(p^\ell) > C$, the above algorithm returns a basis of $\Lambda_\mathbb{Q}$.*

Proof. Assume there exists $i \le j_0$ such that $b_i \notin \Lambda_\mathbb{Q}$. Then there exists a vector $\operatorname{MOD}\big(\Phi(g), p^\ell\big)$ in $\langle b_1, \ldots, b_{j_0}\rangle_\mathbb{Z} \backslash \Lambda_\mathbb{Q}$ not much larger than b_i satisfying the conditions of Lemma 3.46. Let us be more precise: since $\left\|b_{j_0}^*\right\|_2 \le B$, by definition of j_0, we have $\|b_i^*\|_2 \le 2^{(j_0-i)/2}B$ from Siegel condition, and

$$\|b_i\|_2 \le 2^{(i-1)/2}\,\|b_i^*\|_2 \le 2^{(j_0-1)/2}B \le 2^{(n-1)/2}B,$$

where we have used Lemma 2.39. To go from b_i to $\Phi(g)$, we follow the proof of Lemma 3.46,

- we first add $\Phi(\prod f_j)$ for some factor $\prod f_j$ of f, whose length is less than B, so the resulting vector has length $\le (2^{(n-1)/2}+1)B$, which is an upper bound for $|e_i|$,
- we then subtract $e_i\Phi(f_i)$ whose length is less than $|e_i|\,B$.

Let $G = \operatorname{MOD}\big(\phi(g), p^\ell\big) \in \mathbb{Z}[X]$ be the part with degree $< n$. Because of the above, $\|G\|_2 \le (2^{(n-1)/2}+1)B(1+B) =: C$. Because of Lemma 3.46, we have

1. $\operatorname{Res}(G, f) \equiv 0 \pmod{p^\ell}$,
2. $\operatorname{Res}(G, f) \ne 0$,
3. $|\operatorname{Res}(G, f)| < C^n\,\|f\|_2^{n-1} =: R$ is effectively bounded.

The last point is Hadamard's inequality, with $\deg G < n$. If $p^\ell \ge R$, we have a contradiction. Finally, one easily checks that $\log R = O(n^2 + n\log\|f\|_2)$. □

This version of Van Hoeij's Algorithm turns out to have the same proven complexity as LLL. But there are a number of practical improvements, which make it superior in practice:

First we do not work with Λ but with the $\mathbb{Z}$-module generated by the columns of the matrix

$$\begin{pmatrix} I_r & 0 \\ A_1 & B_1 \end{pmatrix},$$

where A_1 is the first line of the $n \times r$ matrix A, and B_1 is the first line of $p^\ell I_n$. We apply LLL to this lattice and eliminate large vectors as above. This gives us a new lattice generated by the columns of a matrix $\begin{pmatrix} M \\ N \end{pmatrix}$. We work now with the lattice generated by by the columns of

$$\begin{pmatrix} M & 0 \\ A_2 & B_2 \end{pmatrix},$$

where A_2 is the second line of the matrix $A \times M$. And so on introducing successively the lines of A (suitably updated from the successive base changes). We expect to eliminate quickly many basis vectors, as if we had worked with all lines of A simultaneously.

A second similar improvement is to take into account only the leading p-adic digits of A (we need to modify the bounds for the discarded vectors of course) then iterate, feeding more and more digits.

Intuitively, we expect to quickly detect true small vectors in the original lattice Λ from these approximate (simpler) lattices: since the small vectors in Λ are small perturbations of the input vectors, they are short vectors in all these other lattices and we hope to detect them early, *before* all the lines and all the digits are input. Theorems exist to support this intuition but are a little harder to formulate and prove!

From these heuristics, the lattices we reduce in practice have much smaller size than the ones in LLL. As a result, van Hoeij's Algorithm is the current best practical method to factor polynomials in $\mathbb{Q}[X]$.

3.3 Factoring in $K[X]$ where K is a number field

Knowing how to factor in $\mathbb{Q}[X]$ allows us to factor over a number field. Let K be a number field of degree n and denote by $\mathcal{O}_K$ the ring of its integers. Let σ_j, $1 \leq j \leq n$, be the n embeddings of K into $\mathbb{C}$. Consider now a polynomial $P \in K[X]$ of degree d,

$$P = \sum_{i=0}^{d} a_i X^i,$$

with $a_d \neq 0$.

As usual we can reduce to a squarefree polynomial by computing its squarefree part $P/\gcd(P, P') \in K[X]$ (here $\mathrm{char} K = 0$), and even to a monic squarefree polynomial. So, from now on, P is monic and squarefree (in particular $a_d = 1$) and we want to factor it in $K[X]$.

Note that the difference with the case $\mathbb{Q}$ is that we cannot reduce to a polynomial with coefficients in $\mathcal{O}_K$ because $\mathcal{O}_K$ is not necessarily an UFD.

Let us extend the σ_j to $K[X]$ by acting on the coefficients and define the norm of $Q \in K[X]$ by

$$\mathcal{N}(Q) = \prod_{i=1}^{n} \sigma_j(Q).$$

By Galois theory we have $\mathcal{N}(Q) \in \mathbb{Q}[X]$ and it is easy to see that if Q is monic, $\mathcal{N}(Q)$ is also monic of degree $n \deg Q$. Furthermore for two polynomials R and S we have $\mathcal{N}(RS) = \mathcal{N}(R)\mathcal{N}(S)$.

Lemma 3.49. *If $Q \in K[X]$ is monic and irreducible, then $\mathcal{N}(P)$ is equal to the power of an irreducible monic polynomial of $\mathbb{Q}[X]$.*

Proof. Let

$$\mathcal{N}(Q) = \prod_{i=1}^{r} T_i^{e_i}$$

be the factorization of $\mathcal{N}(Q)$ into monic irreducible factors in $\mathbb{Q}[X]$. Since $Q \mid \mathcal{N}(Q)$ in $K[X]$ and Q is irreducible in $K[X]$, we have $Q \mid T_i$ in $K[X]$ for some i. But since $T_i \in \mathbb{Q}[X]$ it follows that $\sigma_j(Q) \mid T_i$ for every j and consequently $\mathcal{N}(Q) \mid T_i^n$ in $K[X]$. Our two polynomials being in $\mathbb{Q}[X]$ we have in fact $\mathcal{N}(Q) \mid T_i^n$ in $\mathbb{Q}[X]$ so that $\mathcal{N}(Q) = T_i^m$ for some $m \leq n$ by irreducibility of T_i and because T_i and $\mathcal{N}(Q)$ are monic. □

Suppose now that $K = \mathbb{Q}(\theta)$.

Lemma 3.50. *Let $P \in K[X]$ monic and squarefree. Then there exists only a finite number of $k \in \mathbb{Q}$ such that $\mathcal{N}\big(P(X - k\theta)\big)$ is not squarefree.*

Proof. Let us denote by $\alpha_{i,j}$, $1 \leq i \leq n$, $1 \leq j \leq d$ the roots of $\sigma_i(P)$ in $\mathbb{C}$ and let $k \in \mathbb{Q}$. It is easy to see that $\mathcal{N}(P(X - k\theta))$ is not squarefree if and only if there exist i_1, i_2, j_1, j_2 such that

$$\alpha_{i_1,j_1} + k\sigma_{j_1}(\theta) = \alpha_{i_2,j_2} + k\sigma_{j_2}(\theta),$$

or equivalently

$$k = \frac{\alpha_{i_1,j_1} - \alpha_{i_2,j_2}}{\sigma_{j_2}(\theta) - \sigma_{j_1}(\theta)}.$$

But there are only a finite number of such k. □

Finally we have the following result that will give us an algorithm to factor P.

Theorem 3.51. *Let $Q \in K[X]$ monic and squarefree and assume that $\mathcal{N}(Q)$ is squarefree. Let $\mathcal{N}(Q) = \prod_{i=1}^{r} T_i$ be the factorization of $\mathcal{N}(Q)$ into monic irreducible factors in $\mathbb{Q}[X]$. Then*

$$Q = \prod_{i=1}^{r} \gcd(Q, T_i)$$

is the factorization of Q into monic irreducible factors in $K[X]$ (if we respect the usual condition for the gcd to be monic).

Proof. Let

$$Q = \prod_{i=1}^{s} Q_i$$

be the factorization of Q into monic irreducible factors in $K[X]$.

Since $\mathcal{N}(Q)$ is squarefree, $\mathcal{N}(Q_i)$ is also and by Lemma 3.49 $\mathcal{N}(Q_i) = T_{j(i)}$ for some $j(i)$.

Furthermore, since for $j \neq i$, $\mathcal{N}(Q_i Q_j) \mid \mathcal{N}(Q)$ hence is squarefree, $\mathcal{N}(Q_i)$ is coprime to $\mathcal{N}(Q_j)$.

So by suitable reordering, we obtain $\mathcal{N}(Q_i) = T_i$ and also $r = s$.

Finally, since for $j \neq i$, Q_j is coprime to T_i, and since Q_i is monic, it follows that

$$Q_i = \gcd(Q, T_i).$$

□

This leads to the following algorithm.

Algorithm 14. Factoring in $K[X]$

Input: $K = \mathbb{Q}(\theta)$, $P \in K[X]$ monic and squarefree.

Output: The factorization of P into monic irreducible factors in $K[X]$.

```
1: k=0.
2: if $\mathcal{N}(P(X - k\theta))$ is not square free then
3:     $k = k + 1$ and goto 2.
4: Put $T = \mathcal{N}(P(X - k\theta)) \in \mathbb{Q}[X]$    {at this point $T$ is squarefree.}
5: Factor $T$ in $\mathbb{Q}[X]$: $T = \prod_{i=1}^{r} T_i$, where $T_i$ are monic.
6: for $i$ from 1 to $r$ do
7:     $P_i = \gcd(P, T_i(X + k\theta))$ in $K[X]$ (Euclidean).
8: Return the $P_i$.
```

3.4 Factoring in $\mathbb{C}[X]$

We fix $\varepsilon > 0$ and $P \in \mathbb{Q}(i)[X] \subset \mathbb{C}[X]$, P is monic and of degree n. We want to find $u_1, \cdots, u_n, \in \mathbb{Q}(i)$ such that

$$\left\| P - \prod_{i=1}^{n} (X - u_i) \right\|_1 \leq \varepsilon \left\| P \right\|_1 . \tag{3.4}$$

Theorem 3.52 (Schönhage)**.** *If the coefficients of P are given by floating point numbers, we can compute n floating point numbers $(u_1, \ldots, u_n)$ satisfying* (3.4) *in time*

$$\widetilde{O}(n^3 \ln(s) + n^2 s), \quad \textit{with} \quad \varepsilon = 2^{-s}.$$

Remark 3.53. It is an impressive result, because it does not depend on the *size* of the input polynomial but only on the degree and requested accuracy. We shall not prove the theorem, which is rather technical, but introduce the main ideas.

Remark 3.54. The problem of root finding is ill-conditioned, i.e. sensitive to perturbations (hence difficult), even though the roots are a continuous function of the coefficients (of monic polynomials of given degree). Indeed, let $P = X^n$ and $\widehat{P} = X^n - \varepsilon^n$ for a small $0 < \varepsilon < 1$. The polynomials P and $\widehat{P}$ are very close but ε is a root of $\widehat{P}$ whose distance to the unique root 0 of P is ε, much larger than the distance from P to $\widehat{P}$ $(= \varepsilon^n)$.

Remark 3.55. If $P = \prod_i (X - z_i)$ then $|u_i - z_i|$ will be small (see Ostrowski's Theorem 3.62). The problem in this form is well suited for approximate inputs: instead of estimating the error between the result and the true roots, we estimate the distance from the input to a virtual input which would produce the approximate roots as an exact result. (Backward error analysis.)

3.4.1 Idea of this algorithm

The naive root finding idea is to use random Newton iterations : pick a random $x_0 \in \mathbb{C}$ and for $n \geq 0$, define

$$x_{n+1} = x_n - \frac{P(x_n)}{P'(x_n)}.$$

This sequence will probably converge to a root of P. It works nicely for small degrees but it is quite unstable, especially if the degree of P is big (say, $\deg P \geq 10$) or if $\|P\|_\infty$ is big. Already in degree 3, consider all polynomials in a given ball B, whose roots belong to some other ball Ω. There exists a set of polynomials of positive measure in B, such that a positive measure of initial points in Ω do not yield converging sequences.

Schönhage's idea is deterministic and recursive.

1. Look for a separating circle Γ containing exactly k roots of P: $u_1, u_2, \cdots, u_k$. Ideally, $k \approx n/2$ and the roots of P are well away from Γ (Lemma 3.56).

2. Use Cauchy formula: for all $m \leq k$, approximate the Newton sums
$$s_m := u_1^m + u_2^m + \cdots + u_k^m = \frac{1}{2i\pi} \oint_\Gamma \frac{P'(z)}{P(z)} \cdot z^m \, dz.$$

3. Thanks to Newton formulas, from approximations of the s_m, $m \leq k$, we obtain an approximation P_0 of $\prod_{i=1}^{i=k} (X - u_i)$.

4. Reapply to P_0 and P/P_0.

In order to obtain an efficient algorithm, one must prove perturbation results so as to use floating point arithmetic (incurring rounding errors) with smallest possible accuracy and still realistic error bounds.

In practice, numerical integration is very costly and we alternate these with Newton-like iterations (seeded by the values from the integrals), falling back to integration when the iteration diverges. We will neglect these aspects, see [11] for details.

Then the main problem is to find the separating circle Γ, which ultimately depends on our ability to approximate the modulus of the roots.

3.4.2 Numerical integration

In this section we assume Γ is known. By translation and scaling, we may further assume Γ is the unit circle. We discretize Γ at H-th roots of 1 and approximate the integral by Riemann sums:

Lemma 3.56. *Let*

$$W_m = \frac{1}{H}\sum_{j=0}^{H-1} \frac{P'}{P}(\omega^j)\omega^{(m+1)j}, \quad \omega = e^{2i\pi/H}.$$

Assume P of degree n has no root in the annulus $e^{-\delta} < |z| < e^{\delta}$, for some $\delta > 0$. Then for all $m \leq n < H$, we have

$$|W_m - s_m| \leq \frac{ne^{-\delta(H-m)}}{1-e^{-\delta H}} = O_{n,\delta}(e^{-\delta H}).$$

Proof. Let $u_1, \dots, u_k$ be the roots of P within Γ, and $u_{k+1}, \dots, u_n$ the others. By assumption,

$$|u_1|, \dots, |u_k|, |u_{k+1}|^{-1}, \dots, |u_n|^{-1}$$

are all less than $e^{-\delta} < 1$. Let

$$s_m = \sum_{i\leq k} u_i^m, \quad S_m = \sum_{i>k} u_i^{-m}$$

and develop $(P'/P)(\omega^t)$ in Fourier series,

$$\frac{P'}{P}(\omega^j) = \sum_{\ell\in\mathbb{Z}} c_\ell \omega^{\ell j}.$$

From the logarithmic derivative formula, one obtains

$$\begin{aligned}\frac{P'}{P}(\omega^j) &= \sum_{i\le k}\frac{1}{\omega^j-u_i}+\sum_{k<i}\frac{1}{\omega^j-u_i}\\ &= \sum_{i\le k}\omega^{-j}\sum_{\ell\ge 0}(u_i\omega^{-j})^\ell-\sum_{k<i}u_i^{-1}\sum_{\ell\ge 0}(\omega^j u_i^{-1})^\ell\\ &= \sum_{\ell\ge 0}s_\ell\omega^{-j(\ell+1)}-\sum_{\ell\ge 0}\omega^{j\ell}S_{\ell+1}.\end{aligned}$$

So that

$$c_\ell=\begin{cases}-S_{\ell+1} & \text{if } \ell\ge 0,\\ s_{-\ell-1} & \text{if } \ell<0.\end{cases}$$

From this we gather $W_m=\sum c_\ell$, where $m+1+\ell\equiv 0 \pmod{H}$, and $s_m=c_{-(m+1)}$. Finally, $W_m-s_m=\sum c_\ell$, where ℓ is of the form $-(m+1)+\lambda H$, $\lambda\in\mathbb{Z}\setminus\{0\}$.

From $|s_m|\le ke^{-m\delta}$ and $|S_m|\le(n-k)e^{-m\delta}$, we get

$$\begin{aligned}|W_m-s_m| &\le \sum_{\lambda>0}(n-k)e^{-\delta(\lambda H-m)}+\sum_{\lambda<0}ke^{-\delta(-\lambda H+m)}\\ &\le e^{\delta m}(k+(n-k))\sum_{\lambda>0}e^{-\delta\lambda H}.\end{aligned}$$

□

Remark 3.57. If n is large, to compute the Riemann sums W_m, we choose H a power of 2 and use the FFT to compute the $P(\omega^j)$ and $P'(\omega^j)$ (saving essentially a factor n).

3.4.3 Choosing Γ

Let $(u_1,\dots,u_n)$ be the roots of $P=\sum_{i\le n}a_iX^i$, ordered by increasing modulus, and $\rho_1(P)\le\dots\le\rho_n(P)$ their absolute values. We assume $n\ge 2$. In this section we show that if we can approximate the ρ_i, then we can find a suitable Γ and compute k and δ as above. We always assume that $\rho_1(P)>0$ (since we can first remove the roots equal to 0).

1. Translate: replacing P by $P(X-a_{n-1}/a_n)$, we can assume that the barycenter of the u_i is 0, i.e. that $\sum u_i=0$.

2. Rescale: replacing P by $P(X/\rho_n)$, we can assume that the largest root is close to the unit circle.

3. Test for center: let $S = \{\pm 2, \pm 2i\}$. From Lemma 3.58, there exists a point $\Omega \in S$ such that ρ_n/ρ_1 for $P(X-\Omega)$ is bigger than $2/\left|2 - e^{i\pi/4}\right| \approx 1.35$. Fix the center of Γ at Ω.

4. Choose radius: by translation again, we may now assume that Γ is centered at 0. If $\rho_i < R < \rho i + 1$ is the chosen radius, the optimal δ such that no z in the annulus $Re^{-\delta} < |z| < Re^{\delta}$ is a root of P satisfies
$$e^{\delta} = \min(\rho_{i+1}/R, R/\rho_i).$$
This is maximal for $R = \sqrt{\rho_i \rho_{i+1}}$, with value $e^{\delta} = \sqrt{\rho_{i+1}/\rho_i}$. So we choose k such that ρ_{k+1}/ρ_k is maximal and set the radius R of Γ as above.

   Since the product of the ρ_{i+1}/ρ_i is bigger than 1.35, one of them is $> 1.35^{1/n}$. Finally, $e^{\delta} > 1.35^{1/2n}$ is bounded away from 1.

Lemma 3.58. *With the notations of 3), let $\Omega \in S$ a closest point to a largest root of P. Then ρ_n/ρ_1 for $P(X - \Omega)$ is bigger than $2/\left|2 - e^{i\pi/4}\right|$.*

Proof. There exists a root u of P such that $|\Omega - u| \geq 2$ (since 0 is their center of gravity). There exists a root v of P such that $|\Omega - v| \leq \left|2 - e^{i\pi/4}\right|$ (a closest root). The result follows. □

3.4.4 Graeffe's method (estmate $\rho_k(P)$)

We start with two lemmas, giving very rough bounds, see [12].

Lemma 3.59. *Let $P = \sum_{i \leq n} a_i X^i \in \mathbb{C}[X]$ with $a_0 a_n \neq 0$ and $n \geq 2$. Let $k \geq 0$ such that $|a_k| = \max_i |a_i|$. Then $\rho_k(P) \leq 2n$ and $\rho_{k+1}(P) \geq 1/2n$.*

Proof. The second bound follows from the first applied to the reciprocal polynomial $X^n P(1/X)$ of P. The case $k = 0$ is Cauchy's bound 2.50. The case $k > 0$ follows from Cauchy's bound applied to P/Q, $Q = (X - u_n) \ldots (X - u_{k+1})$. Details left as an exercise (use the series development of $1/Q$). □

Lemma 3.60. *Let $P = \sum_{i \leq n} a_i X^i$, $n \geq 1$. One can choose $r > 0$ such that $Q := P(rX) = \sum_{i \leq n} b_i X^i$ is such that there exists ℓ, h in $[0, n]$, with*
$$\ell < k \leq h, \quad |b_\ell| = |b_h| \geq |b_j|, \quad \forall j \leq n).$$
In particular
$$\frac{1}{2n} \leq \rho_{\ell+1}(Q) \leq \rho_k(Q) \leq \rho_h(Q) \leq 2n.$$

Proof. Let C be the upper convex hull of the $M_j = (j, \log|a_j|) \in \mathbb{R}^2$ (omit the points with $a_j = 0$). Let ℓ be the largest $j < k$ such that $M_j \in C$, and h the smallest $j \geq k$ such that $M_j \in C$. Then set $r = |a_\ell/a_h|^{1/(h-\ell)}$. □

We now use Graeffe's construction to "amplify" the behavior we want to detect: let $G(P)$ be the polynomial G such that $G(X^2) = P(X)P(-X)$ (the letter G stands for Gräffe, the method is also independently due to Dandelin and Lobachevsky). Obviously, G has the same degree as P and its roots are the squares of the roots of P.

Algorithm 15. Graeffe's method for $\rho_k(P)$

```
Input: $P \in \mathbb{C}[X]$, $k \leq \deg P$, $\delta > 0$
Output: $R$ such that $Re^{-\delta} \leq \rho_k(P) \leq Re^{\delta}$.
let $m$ be minimal such that $(2n)^{1/2^m} \leq e^{\delta}$.
Define $P_i$, $Q_i$, $r_i$ by
    $$P_0 = P, \quad Q_j = P_j(r_j X), \quad P_{j+1} = G(Q_j),$$
    where $(Q_j, r_j)$ come from Lemma 3.60.   {The roots of $Q_j$ are the $(u_i/R_j)^{2^j}$, $1 \leq i \leq n$, where $R_j = r_0 r_1^{1/2} \dots r_j^{1/2^j}$.}
Return $R_m$.
```

Proof. By Lemma 3.60, we have $1/2n \leq \rho_k(Q_m) \leq 2n$, hence

$$R_m(2n)^{-1/2^m} \leq \rho_k(P) \leq R_m(2n)^{1/2^m}.$$

□

3.4.5 Continuity of the roots

This is the following result:

Theorem 3.61. *Let $\mathcal{C}$ be the set of monic polynomials of degree n in $\mathbb{C}[X]$, equipped with the natural product topology (we identify $\mathcal{C}$ with $\mathbb{C}^n$). Then $T : \mathbb{C}^n/S_n \to \mathcal{C}$ given by $(z_1, \dots, z_n) \mapsto \prod_{i \leq n}(X - z_i)$ is a homeomorphism.*

Proof. Since $\mathbb{C}$ is algebraically closed (fundamental theorem of algebra), T is a bijection, obviously continuous (polynomial map). Let $S = T^{-1}$ and $P_k \in \mathcal{C}$ be such that $P_k \to P$, we want to prove that $S(P_k) \to S(P)$. But $S(P_k)$ is bounded (Cauchy's bound). By compacity, passing to a subsequence, we may assume that $S(P_k) \to Q$. By continuity of T, we have $P_k = T(S(P_k)) \to T(Q)$, hence $P = T(Q)$ and $Q = S(P)$. □

The following theorem is an effective version, giving concrete approximations to the roots of P from Schönhage's algorithm.

Theorem 3.62 (Ostrowski)**.** *Let $\rho = \max(1, |z_1|, \dots, |z_n|)$. Let $P \in \mathbb{C}[X]$ be monic such that*

$$\|P - (X - z_1)\dots(X - z_n)\|_1 < \varepsilon^n.$$

There exists a permutation $(u_1, \dots, u_n)$ of the complex roots of P such that

$$(1 - 4\varepsilon)\,|u_i - z_i| < 4\rho\varepsilon.$$

Proof. Let z be a root of $\widehat{P} = (X - z_1)\dots(X - z_n)$. Consider the continuous family of polynomials $H_t = tP + (1-t)\widehat{P} = \widehat{P} - t(\widehat{P} - P)$, $t \in [0,1]$. In particular, $H_1 = P$, $H_0 = \widehat{P}$. By continuity of the roots, there exists a continuous path $t \mapsto d_t(z) \in \mathbb{C}$ such that $d_0 = 0$ and $z + d_t$ is a root of H_t for all t. In particular, $u(z) := z + d_1$ is a root of P. (Choose the $d_t(z)$ such that $z \mapsto u(z)$ is 1-to-1.) Let $D = |u - z| = |d_1|$; by continuity of d_t, d_t takes at least all values in $[0, D]$.

On the other hand, we have

$$\left|\widehat{P}(z + d_t)\right| = t\left|(\widehat{P} - P)(z + d_t)\right| < \varepsilon^n(\rho + |d_t|)^n,$$

$$\left|\widehat{P}(z + d_t)\right| \geq \left|\prod_{i=1}^{n} |d_t| - |z - z_i|\right|.$$

Hence, for all $d \in [0, D]$,

$$\varepsilon^n(\rho + D)^n \geq \varepsilon^n(\rho + d)^n > \left|\prod_{i=1}^{n} d - |z - z_i|\right|.$$

By the minimax property of Chebyshev polynomials (Lemma 3.63), the right hand side is larger than $2(D/4)^n$ for some $d \in [0, D]$, whence $\varepsilon(\rho + D) > D/4$ and the result follows. □

For the following minimax result used above, see [5].

Lemma 3.63. *If $P \in \mathbb{R}[X]$ runs through the monic polynomials of degree n, we have*

$$\min_P \max_{x \in [a,b]} |P(x)| \geq \left(\frac{b-a}{2}\right)^n 2^{1-n} = 2\left(\frac{b-a}{4}\right)^n,$$

Remark 3.64. The minimal value is realized by a suitable Chebyshëv polynomial, e.g. if $[a, b] = [-1, 1]$, $2^{1-n}T_n$ (where $T_n(X) = \cos\arccos(nX)$ for $X \in [-1, 1]$).

Chapter 4

Integers

4.1 Elementary algorithms

4.1.1 Introduction

If N is a positive integer, given by a string of binary digits, we consider three basic different problems:

1. (Primality) let N be prime, prove it is so.

2. (Compositeness) let N be composite, prove it is so.

3. (Split) let N be composite, find a non trivial factor $d \neq 1, N$.

In fact, the problem we are actually most interested in would be

4. (Factor) factor N completely, i.e find all prime divisors p of N, the $v_p(N)$, and prove that each p is a prime number.

But an algorithm solving 4. also solves the three others. Conversely, solving 1., 2. and 3. implies we can solve 4.: apply 3. $O(\log N)$ times and run simultaneously 1. and 2. on each factor (stop whenever one succeeds). In theory, 2. is not needed, since 3. is obviously stronger (exhibiting a factor proves compositeness). In practice, ordering the problems by increasing order of difficulty: $2 \ll 1 \ll 3$, and we will consider them in this order. Some landmarks:

Theorem 4.1 (Rabin)**.** *Problem 2. can be solved in randomized time* $\widetilde{O}(\log N)^2$.

Theorem 4.2 (Agrawal-Kayal-Saxena, 2002)**.** *1. and 2. can be solved in polynomial time* $O(\log N)^{10.5}$.

This timing was later improved to $O(\log N)^{6+\varepsilon}$ by Lenstra and Pomerance.

Theorem 4.3 (Miller, 1976)**.** *Assuming the Generalized Riemann Hypothesis holds, 1. and 2. can be solved in time $\widetilde{O}(\log N)^4$.*

Conjecture 4.4 (Goldwasser-Killian, Atkin, Shallit)**.** Fast variants of the ECPP algorithm solve 1. in randomized time $\widetilde{O}(\log N)^4$.

If successful, the Elliptic Curve Primality Proving algorithm (ECPP) produces a primality proof, i.e. a tailor-made algorithm which proves the primality of N in time $\widetilde{O}(\log N)^3$. Something which Miller's algorithm is not capable of.

Definition 4.5. Let $0 \leq \alpha \leq 1$, we define

$$L_\alpha(N) = \exp\big((\log N)^\alpha (\log\log N)^{1-\alpha}\big).$$

Note that $L_0(N) = \log N$ and $L_1(N) = N$.

Theorem 4.6 (Lenstra-Pomerance)**.** *The integer N can be factored in randomized time $L_{1/2}(N)$.*

Conjecture 4.7 (Pollard)**.** Using the Number Field Sieve algorithm (NFS), the integer N is factored in randomized time $L_{1/3}(N)$.

4.1.2 Characters

We introduce here an important notion, which we need to explain the Solovay-Strassen test, and will be the key to Miller's Theorem 4.3.

Definition 4.8. A character χ modulo N is a group homomorphism from the multiplicative group $(\mathbb{Z}/N\mathbb{Z})^*$ to $\mathbb{C}^*$. We lift χ to a function $\chi : \mathbb{Z} \to \mathbb{C}$ by setting $\chi(a) = 0$ if $(a, N) > 1$ and $\chi(a) := \chi(a \mod N)$ otherwise. We write χ_0 for the trivial character: $\chi_0(a) = 1$ whenever $(a, N) = 1$.

An important example is the Jacobi symbol, generalizing the Legendre symbol. For odd $N = \prod p$, where the primes p are repeated according to their multiplicity, let

$$\left(\frac{a}{N}\right) := \prod \left(\frac{a}{p}\right),$$

where $\left(\frac{a}{p}\right)$ is Legendre's symbol (-1, 0, or 1 if a is a non-square, 0 or a non-zero square in $\mathbb{F}_p$).

The function $a \mapsto \left(\frac{a}{N}\right)$ is a character modulo N, whose values can be computed efficiently using the quadratic reciprocity law, and a variant of the Euclidean algorithm, in time $\widetilde{O}(\log N)^2$ if we use fast arithmetic. If N is prime, then $\left(\frac{a}{N}\right) \equiv a^{(N-1)/2} \pmod{N}$.

4.1.3 Compositeness

The following congruences are the basis of three historically important compositeness tests:

Theorem 4.9. *If $N > 2$ is prime and $0 < a < N$, then the following equalities hold in $\mathbb{Z}/N\mathbb{Z}$:*

1. $a^{N-1} = 1$,
2. $a^{(N-1)/2} = \left(\frac{a}{N}\right) \neq 0$,
3. *Write $N - 1 = 2^e q$, q odd, $e \geq 1$ and set $b := a^q$. Then $b = 1$ or there exists a unique $0 \leq i < e$ such that $b^{2^i} = -1$.*

Proof. (1) is Fermat's little theorem and (2) is one of the elementary properties of the Legendre symbol. The condition $\left(\frac{a}{N}\right) \neq 0$ is empty if N is prime, but we shall need it for general N later.

The last congruence follows from the polynomial equality

$$X^{2^e} - 1 = (X - 1)(X + 1)(X^2 + 1) \dots (X^{2^{e-1}} + 1)$$

(which you can prove directly or as a property of cyclotomic polynomials), evaluated at $X = b$: the left hand side is 0 by Fermat so one of the factors on the right hand side is 0. The index i is obviously unique since from then on, the b^{2^j}, $j > i$ are obtained by successive squarings, hence all are equal to 1.

This yields a more enlightening proof: by Fermat, $a^{N-1} = 1 = b^{2^e}$. In a (commutative) field of characteristic different from 2, the equation $X^2 = 1$ has just two solutions -1 and 1. Since we obtain 1 by a sequence of squarings from b, either all b^{2^i} are 1 or we hit -1 somewhere. □

These congruences yield three probabilistic compositeness tests: pick a random $0 < a < N$, and check one of (1), (2) or (3) above, yielding respectively Fermat's test (1640), Solovay-Strassen test (1977), and Rabin-Miller test. As for the last test, Miller (1976) devised a conditional deterministic test (see Theorem 4.3), which Rabin modified to the above probabilistic compositeness test in 1980. For completeness:

Algorithm 16. Fermat compositeness test

Input: N an odd integer, $a \in \mathbb{Z}/N\mathbb{Z}$, $a \neq 0$.

Output: $F(a) =$ `Composite` or `Fail`

```
1: If $a^{N-1} = 1$ return Fail.
2: Return Composite.
```

Algorithm 17. Solovay-Strassen compositeness test

```
Input: $N$ an odd integer, $a \in \mathbb{Z}/N\mathbb{Z}$, $a \neq 0$.
Output: $SS(a) =$ Composite or Fail
If $\left(\frac{a}{N}\right) = a^{(N-1)/2} \neq 0$ return Fail.
Return Composite.
```

Algorithm 18. Rabin-Miller compositeness test

```
Input: $N$ an odd integer, $a \in \mathbb{Z}/N\mathbb{Z}$, $a \neq 0$.
Output: $RM(a) =$ Composite or Fail
Write $N - 1 = 2^e q$, $q$ odd.
Compute $b = a^q$.
If $b = 1$ or $b^{2^i} = -1$ for some $i = 0, \dots, e-1$ return Fail.
Return Composite.
```

Theorem 4.10. *All three tests (Fermat, Solovay-Strassen, Rabin-Miller) run in time* $\widetilde{O}(\log N)^2$.

Proof. This is obvious for Fermat, easy for Rabin-Miller (note that we need $O(\log q + e) = O(\log N)$ multiplications in $\mathbb{Z}/N\mathbb{Z}$), and a little more involved for Solovay-Strassen: the powering part is easy again, but one needs to adapt the complexity proof of Euclid's algorithm to the computation of Jacobi's symbol $\left(\frac{a}{N}\right)$ using quadratic reciprocity. This requires specifying precisely the latter, and you can check out the result in Cohen. □

Definition 4.11. An a such that $SS(a)$ or $RM(a)$ returns `Composite` is called a *witness* (of compositeness). If N is composite, an a such that they return `Fail` is called a *liar*.

Note that a witness testifies to the compositeness of N, but provides only circonstancial evidence: no explicit factor is exhibited.

Theorem 4.12. *Let* $N > 2$ *be an odd integer (no longer a prime),* $0 < a < N$*, and consider again the congruences from Theorem 4.9. Then (3) implies (2) implies (1).*

Proof. (1) follows from (2) by squaring: since we impose $\left(\frac{a}{N}\right) \neq 0$ in (2), the Jacobi symbol is ± 1. We now prove (3) ⇒ (2), which is surprisingly intricate. Note that (3) ⇒ (1) is obvious again by squarings, which already implies $a \in (\mathbb{Z}/N\mathbb{Z})^*$ and $\left(\frac{a}{N}\right) = \pm 1$. Since q is odd, we have $\left(\frac{a}{N}\right) = \left(\frac{a}{N}\right)^q = \left(\frac{b}{N}\right)$.

1. If (3) holds because $b = 1$, then $a^{(N-1)/2} = b^{2^{e-1}} = 1$ and $\left(\frac{b}{N}\right) = 1 = \left(\frac{a}{N}\right)$, hence (2) holds.

2. We now assume that $b^{2^i} = -1$ for a unique $0 \leq i < e$. We write $p - 1 = 2^{e_p} q_p$, q_p odd, for all $p \mid N$. The congruence $b^{2^i} \equiv -1 \pmod{p}$ holds for all $p \mid N$, meaning that the order of $b \in \mathbb{F}_p^*$ is 2^{i+1}. This must divide $\sharp\, \mathbb{F}_p^* = p - 1$, hence $i + 1 \leq e_p$.

3. From $\left(\frac{b}{p}\right) \equiv b^{2^{e_p - 1} q_p} \pmod{p}$, $b^{2^i} \equiv -1 \pmod{p}$ and $i + 1 \leq e_p$, it follows that $(-1)^{2^{e_p - i - 1}} = \left(\frac{b}{p}\right)$ (we also use that q_p odd). Hence $\left(\frac{b}{p}\right) = -1$ if and only if $e_p = i + 1$.

4. Consider
$$\begin{aligned} N = \prod_p (1 + 2^{e_p} q_p) &\equiv 1 + 2^{i+1} \sum_{p\colon \left(\frac{b}{p}\right) = -1} q_p \pmod{2^{i+2}} \\ &\equiv 1 + 2^{i+1} \# \left\{ p\colon \left(\frac{b}{p}\right) = -1 \right\} \pmod{2^{i+2}}, \end{aligned}$$
where each p is repeated according to its multiplicity. It follows that $i + 1 = e$ if and only if $\# \left\{ p\colon \left(\frac{b}{p}\right) = -1 \right\}$ is odd, that is if and only if $\left(\frac{b}{N}\right) = -1$.

5. Finally $a^{(N-1)/2} = b^{2^{e-1}} = -1$ if and only if $i = e - 1$.

In other words $a^{(N-1)/2}$ and $\left(\frac{a}{N}\right)$, which are both equal to ±1, are equal to -1 under the exact same conditions. Thus they are equal. □

In other words, Rabin-Miller's test is stronger than Solovay-Strassen, itself stronger than Fermat. It is easy to find specific counter examples showing that the reverse implications do not hold in general. To check whether our tests are any good we must now prove that for any odd composite N, there exist sufficiently many $0 < a < N$ that violate (1), (2) or (3).

Theorem 4.13. *If $N > 1$ is composite, then*
$$\left\{ 0 < a < N \colon a^{N-1} \equiv 1 \pmod{N} \right\}$$
is a subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$. It is equal to the full group $(\mathbb{Z}/N\mathbb{Z})^*$ if and only if N is squarefree and $p \mid N \Rightarrow p - 1 \mid N - 1$ for all prime divisors p of N.

Proof. The subgroup assertion is clear. Now assume $a^{N-1} \equiv 1 \bmod N$ for all a coprime to N. First, if $p^e \mid N$ with p prime and $e \geq 2$, by the Chinese Remainder Theorem there exists $a \in \mathbb{Z}$ such that $a \equiv 1 + p^{e-1} \bmod p^e$ and $a \equiv 1 \bmod N/p^e$. Then a has order p mod p^e hence mod N. But since

$(a, N) = 1$ we have $a^{N-1} \equiv 1 \bmod N$ so that $p \mid N-1$, a contradiction. This proves that N is squarefree. Now let p be a prime divisor of N and ω a primitive root mod p. Since N is squarefree the CRT shows that there exists $a \in \mathbb{Z}$ with $(a, N) = 1$ such that $a \equiv \omega \bmod p$ and $a \equiv 1 \bmod N/p$. Such an a has order $p-1 \bmod p$ hence mod N, and since $a^{N-1} \equiv 1 \bmod N$, we have $p-1 \mid N-1$.

Conversely, suppose N is squarefree and $p-1 \mid N-1$ for every prime divisor p of N. Consider a coprime to N and a prime divisor p of N. Then a is coprime to p and $a^{p-1} \equiv 1 \bmod p$. But $p-1 \mid N-1$ so that $a^{N-1} \equiv 1 \bmod p$. It follows from the CRT that $a^{N-1} \equiv 1 \bmod N$. □

A bad composite N as above, which completely wrecks Fermat's test, is called a Carmichael number. They were actually previously defined and studied by Korselt (1899), who could not find an actual example. In 1910, Carmichael found the smallest such number 561.

Corollary 4.14. *A Carmichael number is odd and has at least three distinct primes in its decomposition.*

Proof. First if $N = 2\prod_{i=1}^{r} p_i$ with p_i odd and $r \geq 1$ (N is composite) it is impossible to have $p_1 - 1 \mid N-1$ because $p_1 - 1$ is even and $N-1$ is odd. Suppose now that $N = pq$ with $p \neq q$ odd. We have $p-1 \mid pq-q$ and $p-1 \mid N-1 = pq-1$ so that $p-1 \mid q-1$. By symmetry $p-1 = q-1$ and N is not squarefree. □

Remark 4.15. Another definition of a Carmichael number could be : N is a Carmichael number if and only if

$$a^N \equiv a \quad \text{for every } 0 < a < N.$$

In fact, this implies obviously our definition. Conversely suppose that N is a Carmichael number and take an integer a. For every prime divisor p of N, we have $a \equiv 0 \bmod p$ or a coprime to p, so that $a^N \equiv a \bmod p$ in each case (see above for the second case). The CRT gives the conclusion.

The Carmichael numbers have no equivalent for the Solovay-Strassen test, nor for the stronger Rabin-Miller test:

Theorem 4.16. *If $N > 1$ is composite, then*

$$\left\{0 < a < N \colon a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \neq 0 \pmod{N}\right\}$$

is a proper subgroup of $(\mathbb{Z}/N\mathbb{Z})^*$.

Proof. The subgroup assertion is clear again, and we must prove it is not the whole group. Since it is a subgroup of the one considered in the previous theorem, we may assume that N is a Carmichael number, and we must find an a not belonging to the given set.

If $p \mid N$ is an odd prime, let a be a non-square mod p and $a \equiv 1 \pmod{N/p}$, which exists by the Chinese Remainder Theorem. In fact, since N is squarefree, we have $\gcd(p, N/p) = 1$. This implies $\left(\frac{a}{N}\right) = -1$, but $a^{(N-1)/2} \not\equiv -1 \pmod N$ since this is not true modulo N/p. □

Corollary 4.17. *If N is an odd composite number, an $0 < a < N$ chosen uniformly at random is a witness with probability $\geq 1/2$.*

Proof. A proper subgroup has index greater than 2. □

This is a quick and beautiful argument, telling us witnesses are relatively abundant. In fact for most N almost all a are witnesses. It is not hard to give a much more precise result, or to improve on the Corollary:

Theorem 4.18. *Let $N > 2$ be an odd integer.*

1. *If $N = \prod p^{f_p}$, let $\omega = \#\{p \mid N\}$,*

$$e_p = v_2(p-1), \quad E = \min_{p|N} e_p \geq 1.$$

Then the number of a such that $RM(a)$ returns `Fail` *is*

$$\prod_{p|N} \gcd(q, p-1)\left(1 + \frac{2^{E\omega} - 1}{2^{\omega} - 1}\right).$$

2. *If N is composite, an $0 < a < N$ chosen uniformly at random is a witness with probability $\geq 3/4$.*

Proof. An a such that $RM(a)$ fails belongs to $(\mathbb{Z}/N\mathbb{Z})^*$. By the Chinese Remainder Theorem, $(\mathbb{Z}/N\mathbb{Z})^*$ is ring-isomorphic to $\prod_{p|N}(\mathbb{Z}/p^{f_p}\mathbb{Z})^*$. Each factor is cyclic, which linearizes the equations. We can count the (a_p) in the product $\prod_{p|N} \mathbb{Z}/(p-1)p^{f_p-1}\mathbb{Z}$ such that $RM(a)$ returns `Fail`:

$$\#\{(a_p) : qa_p = 0\} + \sum_{i=0}^{e-1} \#\left\{(a_p) : q2^i a_p = (p-1)p^{f_p-1}/2\right\}$$

(these sets are disjoints). Using $(q, (p-1)p^{f-1}) = (q, p-1)$ (because $(q, N) = (q, p) = 1$), then using the fact that $\alpha x \equiv \beta \bmod \gamma$ has zero solution if

$\beta \nmid \gcd(\alpha,\gamma)$ and $\gcd(\alpha,\gamma)$ solutions modulo γ otherwise, and finally using $E \le e$ (obvious), we see that on the one hand

$$\#\left\{(a_p) : qa_p = 0\right\} = \prod_p \gcd(q, p-1),$$

and that on the other hand

$$\#\left\{(a_p) : q2^i a_p = (p-1)p^{f_p-1}/2\right\} = 0$$

if $i \ge e_p$ for some p, i.e. if $i \ge E$, so that we have

$$\begin{aligned}\sum_{i=0}^{e-1} \#\left\{(a_p) : q2^i a_p = (p-1)p^{f_p-1}/2\right\} &= \sum_{i=0}^{E-1}\prod_p \gcd(2^i q, (p-1)p^{f_p-1}) \\ &= \sum_{i=0}^{E-1}\prod_p 2^i \gcd(q,(p-1)) \\ &= \sum_{i=0}^{E-1} 2^{i\omega}\prod_p \gcd(q,(p-1))\end{aligned}$$

Finally we obtain

$$\left(1+\sum_{i=0}^{E-1}2^{i\omega}\right)\prod_p (q,p-1) = \left(1+\frac{2^{\omega E}-1}{2^\omega-1}\right)\prod_p (q,p-1)$$

and the first result follows. Note that the special case $\omega = 1$, $p = N$, $E = e$ reproves case (3) in Theorem 4.9. We have already seen two rather more enlightening proofs.

To prove the second point, note that the proportion of liars is

$$\frac{\prod_{p|N}\gcd(q,p-1)}{N-1}\left(1+\frac{2^{\omega E}-1}{2^\omega-1}\right).$$

We bound $\gcd(q,p-1) \le (p-1)/2^{e_p} \le (p-1)/2^E$. We treat first the case $\omega = 1$: $N = p^f$, $f > 1$. Then $(p-1)/(p^f-1) \le 1/(p+1) \le 1/4$, and the result follows.

If $\omega > 1$, we bound $\prod(p-1) \le N-1$ and the proportion is less than

$$2^{-\omega E}\left(1+\frac{2^{\omega E}-1}{2^\omega-1}\right).$$

For a fixed ω, this is a decreasing function of E, which equals $2^{1-\omega} \leq 1/4$ for $E = 1$ and $\omega > 2$. Hence the result unless $\omega = 2$, in which case we obtain only $1/2$.

Suppose now that $\omega = 2$. Coming back to the original $\gcd(q, p-1) \leq (p-1)/2^{e_p} \leq (p-1)/2^E$, we see that our final upper bound is divided by at least 2 unless we have equality throughout. The only remaining case is $e_1 = e_2 = E = 1$. Put $N = p_1^{\alpha_1} p_2^{\alpha_2}$, $p_1 = 2q_1 + 1$, $p_2 = 2q_2 + 1$ with q_1 and q_2 odd. Our proportion of liars is

$$\frac{\prod_{p|N} \gcd(q, p-1)}{N-1} \left(1 + \frac{2^{\omega E} - 1}{2^{\omega} - 1}\right) = 2\frac{\gcd(q, p_1 - 1)\gcd(q, p_2 - 1)}{p_1^{\alpha_1} p_2^{\alpha_2} - 1}$$

and we have to prove

$$\frac{\gcd(q, p_1 - 1)\gcd(q, p_2 - 1)}{p_1^{\alpha_1} p_2^{\alpha_2} - 1} \leq \frac{1}{8}.$$

If $\alpha_i \geq 2$ for some i,

$$p_1^{\alpha_1} p_2^{\alpha_2} - 1 \geq 3(p_1 p_2 - 1) \geq 3(4q_1 q_2 + 2q_1 + 2q_2)$$

and $\gcd(q, p_i - 1) = \gcd(q, 2q_i) = \gcd(q, q_i) \leq q_i$ so that our fraction is less than $1/12$. Finally we are reduced to the case $N = p_1 p_2$, where the p_i are distinct primes. Looking back to

$$\frac{\gcd(q, p_1 - 1)\gcd(q, p_2 - 1)}{p_1^{\alpha_1} p_2^{\alpha_2} - 1} = \frac{\gcd(q, q_1)\gcd(q, q_2)}{4q_1 q_2 + 2q_1 + 2q_2},$$

we see that this can be $> 1/8$ only if $\gcd(q, q_i) = q_i$ for $i = 1, 2$ which leads to $p_i - 1 = 2q_i \mid 2q \mid p_1 p_2 - 1$ for $i = 1, 2$. But we have already seen that this is impossible (see Corollary 4.14). □

Definition 4.19. Let $w(N)$ be the least witness for the compositeness of N, i.e. the smallest a such that $RM(a)$ returns `Composite`. For N prime, we let $w(N) = 0$.

If N is a Carmichael number whose prime divisors are $\equiv 3 \pmod 4$, there are exactly $\phi(N)2^{1-\omega}$ liars and $3/4$ is not that far off. But, in general, the lower bound $3/4$ is very pessimistic: for large random N, we expect that $(q, p-1)$ is small for $p \mid N$; then the proportion of liars is essentially bounded by $2^{(\omega-1)e}/q$ which is small. Precisely, we have the following very strong *average* result:

Theorem 4.20 (Burthe)**.** *The average value of $w(N)$ over odd integers is 2:*

$$\sum_{N<X,\, 2\nmid N} w(N) \sim 2 \sum_{N<X,\, 2\nmid N} 1.$$

Which means that 2 is almost always a reliable witness. Note that from the prime number theorem, the primes contribute a negligible amount $O(X/\log X)$ to the right hand side $\sim X$. On the other hand, it is known by work of Alford-Granville-Pomerance that there are infinitely many Carmichael numbers and that $\limsup_{N\to\infty} w(N) = +\infty$.

4.1.4 Primality

Miller's theorem

First we explain how the GRH can turn either Solovay-Strassen or the Miller-Rabin test into a good primality prover. How does the Riemann Hypothesis come into play ? For χ a character modulo N, we introduce the Dirichlet L-function

$$L(\chi, s) = \sum_{n\geq 1} \chi(n) n^{-s}.$$

These are instrumental in proving that there exist infinitely many primes such that $p \equiv a \pmod N$ whenever $\gcd(a, N) = 1$. Just as the Riemann zeta function (which we essentially recover when $\chi = \chi_0$) is required to prove the prime number theorem.

For details on the basic theory of L-functions, see Serre [20]. For more advanced material, see Iwaniec-Kowalski [13]. In particular, it is a standard fact that $L(s, \chi)$ extends to a meromorphic function on $\mathbb{C}$ with at most a simple pole in $s = 1$ (if and only if $\chi = \chi_0$). For $\mathrm{Re}(s) > 1$, it satisfies an Euler product

$$L(s, \chi) = \prod_p (1 - \chi(p) p^{-s})^{-1}.$$

The theorem of Miller, in an effective version due to Bach [3], is as follows:

Theorem 4.21. *Assume the Dirichlet L-functions $L(s, \chi)$ have no 0 in the half-plane $\mathrm{Re}(s) > 1/2$, for all characters modulo N (GRH). If N is composite, there exists a witness (for SS and RM) $a \leq 2(\log N)^2$. In particular $w(N) \leq 2(\log N)^2$.*

Since $RM(a)$ runs in time $\widetilde{O}(\log N)^2$, this theorem yields a conditional primality test in essentially quartic time $\widetilde{O}(\log N)^4$. Unconditionally, we only know that $w(N) \leq N^{1/(6\sqrt{e})+\varepsilon}$ for all $\varepsilon > 0$, and N large. Not sufficient for a polynomial-time test.

Proof. (rough idea). Let $G = (\mathbb{Z}/N\mathbb{Z})^*$. Since N is composite, the subgroup $H = \left\{a \in G\colon a^{(N-1)/2} = \left(\frac{a}{N}\right)\right\}$ is proper. In particular, there exists a non trivial character χ modulo N such that χ is trivial on H (lift a non-trivial character of G/H). Assume by contradiction that $w(N) > x$, then $\sum_{a \leq x} \chi(a) = \lfloor x \rfloor$, and we have a non-trivial character masquerading χ_0. Assuming GRH, we have good bound on character sums, which yields a contradiction.

More precisely, taking the logarithmic derivative of the Euler product, we obtain

$$-\frac{L'}{L}(s,\chi) = \sum_{n \geq 1} \chi(n)\Lambda(n)n^{-s},$$

where Λ is the Riemann-von Mangoldt function: $\Lambda(n) = \log p$ if $n = p^k$ is a prime power, and $\Lambda(n) = 0$ otherwise.

Perron's formula says that

$$\sum_{n \leq x} \chi(n)\Lambda(n) = \frac{1}{2i\pi} \int_{\operatorname{Re}(s)=c} -\frac{L'}{L}(s,\chi)x^s \, \frac{ds}{s},$$

for $x > 0$ not in $\mathbb{Z}$ and any $c > 1$. Subtracting the formula for χ_0, we obtain

$$\int_{\operatorname{Re}(s)=c} \frac{L'}{L}(s,\chi)x^s \, \frac{ds}{s} = \int_{\operatorname{Re}(s)=c} \frac{L'}{L}(s,\chi_0)x^s \, \frac{ds}{s}.$$

We move the line of integration to the left up to the left of the line $\operatorname{Re}(s) = 1/2$: the GRH enables us to keep a tight control on the singularities of L'/L: at the zeroes of L for $\operatorname{Re}(s) = 1/2$, and a simple pole at $s = 1$ for $(L'/L)(s, \chi_0)$. Keeping track of the residues, we find an equality of the form $O(x^{1/2}) = x + O(x^{1/2})$, a contradiction if x is large.

The true proof is technically more demanding because the error terms depend on N: we must use an "explicit formula", and integrate with respect to more involved kernel functions. □

Primality certificates (Pratt)

Besides being conditional, Miller's algorithm is not entirely satisfactory: if the answer is no, I have a witness, i.e. a proof of compositeness; but if the answer is yes, I am left with no evidence, hence no convincing argument besides "I did program the test correctly". We are no better off than when trial dividing up to the square root of the number. In this section, we explain the idea of *primality certificate*, or succinct proof of primality.

We use the following idea: $N > 1$ is prime if and only if $(\mathbb{Z}/N\mathbb{Z})^*$ is cyclic of order $N-1$. If I can exhibit a generator and prove that it has order $N-1$, I am done. Unfortunately, this will require factoring $N-1$, which is hard. But the person I am handing the certificate to will not care: creating a proof may be hard, but checking it is easy.

Theorem 4.22 (Pocklington)**.** *Let $N > 1$ be an integer and $p \mid N-1$ a prime such that $v_p(N-1) = e$. Assume $a \in \mathbb{Z}$ satisfies*

- $a^{N-1} \equiv 1 \pmod{N}$,
- $\gcd(a^{(N-1)/p} - 1, N) = 1$.

Then all divisors $d \mid N$ satisfy $d \equiv 1 \pmod{p^e}$.

Proof. We may assume that d is prime, since a product of integers $\equiv 1 \pmod{M}$ is $\equiv 1 \pmod{M}$. Since $d \mid N$, $a^{N-1} \equiv 1 \pmod{d}$, which implies that $\gcd(a, d) = 1$ hence $a^{d-1} \equiv 1 \pmod{d}$ by Fermat's little theorem.

Since $a^{(N-1)/p} \not\equiv 1 \pmod{d}$, the order r of a in $(\mathbb{Z}/d\mathbb{Z})^*$ satisfies $r \mid N-1$, but $r \nmid (N-1)/p$, hence $p^e \mid r$. On the other hand, $r \mid d-1$ and we are done. □

Corollary 4.23. *Let $N > 1$ be an integer. Assume $N-1 = FU$ with $F \geq \sqrt{N}$, where the prime divisors of F are known, and for all such p, an a_p as above is given. Then N is prime.*

Proof. If $d \mid N$, then $d \equiv 1 \pmod{F}$ by the Chinese remainder Theorem and Theorem 4.22. Hence $d = 1$ or $d \geq F + 1 > \sqrt{N}$. The latter implied $d = N$, since $N/d < \sqrt{N}$ must be 1. Hence N is prime. □

Of course, the catch is that one needs to factor $N-1$. But when this is done, the rest is easy: if N is prime, all $a \in (\mathbb{Z}/N\mathbb{Z})^*$ satisfy the first condition, and exactly $(N-1)/p$ elements satisfy $a^{(N-1)/p} = 1$, hence a random $a \in (\mathbb{Z}/N\mathbb{Z})$ is a suitable a_p with probability $1 - 1/p \geq 1/2$.

We can now define recursively a primality certificate $C(N)$: it is a factorization $N - 1 = \prod p^{e_p}$ together with a set of triples $(p, a_p, C(p))$, where $C(p)$ recursively certifies p. In order to avoid infinite recursion, we allow the empty certificate for $p = 2$.

Exercice 4.24. Bound the size of $C(N)$. How fast can you check it ?

4.1.5 Producing primes

We recall the Prime Number Theorem in its standard form:

Theorem 4.25 (Prime Number Theorem (PNT)). *As $x \to +\infty$, we have*

$$\pi(x) := \#\{p \leq x \colon p \text{ prime}\} \sim \frac{x}{\log x}.$$

Effectively, we have

$$\frac{x}{\log x}\Big(1 + \frac{1}{2\log x}\Big) \leq \pi(x) \leq \frac{x}{\log x}\Big(1 + \frac{3}{2\log x}\Big),$$

where the left-hand side is valid for $x \geq 59$, and the right-hand side for $x > 1$.

From the prime number theorem, the number of primes in $]N, 2N]$ is

$$\pi(2N) - \pi(N) \sim \frac{N}{\log N}.$$

Picking an integer at random in the interval, the expected number of trials before hitting a prime is $\log N$. We can test the numbers produced for compositeness, then for primality once we have a good candidate (an integer which fails many compositeness tests is often declared a "probable prime"). This is a suitable algorithm to find a *big* prime. We now examine a dual, nicer construction: a sieve constructs simultaneously many *small* primes, for an essentially constant unit cost. According to the divide-and-conquer principle, "small primes" algorithms followed by chinese remaindering (and to a lesser degree p-adic algorithms using Hensel lifting) perform better than "large prime" algorithms, although the latter are conceptually simpler.

Algorithm 19. Eratosthenes's sieve

Input: An integer B.

Output: The set of primes $p \leq B$.

```
Initialize an array $A[2] = \cdots = A[B] = 1$.
for $n = 2, \dots, \sqrt{B}$ do
  if $A[n] = 1$ then    {$n$ is prime}
    for $k = 2, \dots, B/n$ do    {cross out multiples of $n$}
      Set $A[kn] = 0$,    {$kn$ not a prime}
Return the $n$ such that $A[n] = 1$.
```

The number of array operations involved is

$$2B + \sum_{p \leq \sqrt{B}} \lfloor B/p \rfloor = 2B + O(\sqrt{B}) + B \sum_{p \leq \sqrt{B}} \frac{1}{p} \sim B \log\log B,$$

using $\sum_{p \leq x} p^{-1} \sim \log\log x$, which follows for instance from the Prime Number Theorem.

4.1.6 Split

This section introduces the important notion of smooth integers, and serves as a warm up for our later factoring with elliptic curves.

Definition 4.26. Let $B > 0$. A positive integer $N > 0$ is

- B-smooth (or B-friable) if $p \mid N$ implies $p \leq B$ for prime p.
- B-powersmooth if $p^k \mid N$ implies $p^k \leq B$ for prime p and positive k.

Theorem 4.27 (de Bruijn)**.** *Let $\psi(x, y) = \#\{n \leq x \colon n \textit{ is } y\textit{-smooth}\}$. Provided $(\log x)^{\varepsilon} \leq u \leq (\log x)^{1-\varepsilon}$ for some $0 < \varepsilon < 1$, we have*

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

as x tends to infinity.

We will abuse this theorem to estimate the probability that integers in certain sequences are $x^{1/u}$ smooth. Our reasoning will not be rigorous: a random integer in our sequence is considered to be $x^{1/u}$-smooth with the same probability u^{-u} as if it were taken form a uniform distribution. It is possible to rigorously prove some estimates, using much more technical arguments, beyond the scope of our lectures.

Here is a simple application of smooth numbers: the Pollard's $p - 1$ method. The idea behind this algorithm is as follows. Let p be a prime (unknown) dividing N. Let $1 < a < N$ an integer that we can assume coprime to N by computing a gcd (if not N will have split). By Fermat's theorem we have $a^{p-1} \equiv 1 \bmod p$. Now assume that $p-1$ is B-powersmooth for a certain B not too large. Then $p-1$ divides the lcm of of the numbers from 1 to B. Hence $a^{\operatorname{lcm}(1,2,\ldots,B)} \equiv 1 \bmod p$ which implies

$$\gcd(a^{\operatorname{lcm}(1,2,\ldots,B)} - 1, N) > 1.$$

Finally we are done if this gcd is not equal to N. This leads to the following algorithm.

Algorithm 20. Pollard's $p-1$ method

Input: N an integer, B a smoothness bound.

Output: A non-trivial factor of N or `Fail`.

```
Using Eratosthenes's sieve, compute all primes $p \leq B$.
Pick a random $a \in \mathbb{Z}/N\mathbb{Z}$. Let $b = a$.
for $p \leq B$ do    {compute $b = a^{\mathrm{lcm}(2,\ldots,B)}$}
    Let $k$ be maximal such that $p^k \leq B$.
    Set $b := b^{p^k}$.
if $d = \gcd(b-1, N)$ is a non-trivial divisor of $N$ then
    Return $d$.
else
    Return Fail.
```

The algorithm succeeds when the order of a modulo some divisor of N is B-powersmooth. Equivalently, the order of a modulo a *prime* divisor p of N is B-powersmooth. Unless we are lucky this means that $p-1$ is B-powersmooth. Indeed, if $\ell > B$ is a large prime dividing $p-1$, then a random $a \in (\mathbb{Z}/N\mathbb{Z})^*$ has order divisible by ℓ in $\mathbb{F}_p^*$ with probability $1 - 1/\ell \approx 1$. ($p-1$ also fails to be B-powersmooth if $\ell^k > B$ divides $p-1$ for a small ℓ and huge k, but this is implausible.)

At the end of the **for** loop, $b = a^{\mathrm{lcm}(2,\ldots,B)}$, which is congruent to 1 modulo p by our assumption that $p-1$ is B-powersmooth. In other words, $p \mid b-1$. If we are lucky, $N \nmid b-1$ and d is a non-trivial factor. The algorithm can fail for two reasons:

- $d = 1$: this proves that there is no prime divisor p of N such that $p-1$ is B-powersmooth. We must increase B and retry.

- $d = N$: the order of a in $(\mathbb{Z}/N\mathbb{Z})^*$ is B-powersmooth. As above, unless we were lucky, this means that $\phi(N)$ is B-powersmooth. We must decrease B and retry. (If B is not very large, we can try a different a, just in case.)

The cost of the method is dominated by the powering, in time

$$\sum_{p^k \leq B} (\log p^k) \widetilde{O}(\log N) = \widetilde{O}(B \log N).$$

Assuming that $p-1$ is B-powersmooth is a little extreme. We increase our chances by assuming it is B_1-powersmooth, up to a single prime less than $B_2 \gg B_1$. There is a nice way of implementing this idea, based on the fact that primes are relatively plentiful, and the differences $p_{i+1} - p_i$ between consecutive primes are rather small. (To be taken with a grain of salt: it is an easy exercise to show that $\limsup_i p_{i+1} - p_i = +\infty$; the "converse" statement $\liminf_i p_{i+1} - p_i = 2$ is the famous *Twin primes conjecture.*)

Algorithm 21. Pollard's $p-1$ method, with B_2 phase

```
Input: $N$ an integer, $(B_1, B_2)$ smoothness bounds.
Output: A non-trivial factor of $N$ or Fail.
Pick a random $a \in \mathbb{Z}/N\mathbb{Z}$ and compute $b = a^{\mathrm{lcm}(2,\dots,B_1)}$ as in the standard method.
Let $S = \{b^{p_{i+1}-p_i}\}$, where the $p_i$, $1 \le i \le K$ are the consecutive primes $B_1 < p \le B_2$. This is a (small) set indexed by the $p_{i+1} - p_i$.
Set $b \leftarrow b_1^p$;
for $i = 1, \dots, K-1$ do
    If $d = \gcd(b-1, N)$ is a non-trivial divisor of $N$, return $d$.
    Replace $b \leftarrow b \times S[p_{i+1} - p_i]$.    $\{b = a^{\mathrm{lcm}(2,\dots B_1)p_{i+1}}\}$
Return Fail.
```

In effect, instead of a powering cost of $\sum_{p<B_2} \log p \sim B_2$ multiplications, we have $\pi(B_2) \sim B_2/\log B_2$ multiplications. If B_2 is so large that a full fledged sieving of $[0, B_2]$ becomes impractical, we can precompute the primes up to $\sqrt{B_2}$ and sieve out smaller slices $[kM, (k+1)M]$ with $k = \lfloor B_2/M \rfloor$.

The $p-1$ method has the very nice feature of being sensitive to the size of the smallest prime divisor of N: smaller numbers are smoother. On the other hand the first point above is a big problem: either there exists a prime divisor p with $p-1$ smooth, or there does not and we are lost. The elliptic curve method nowadays completely supersedes $p-1$: it will try *many* orders $\#E(\mathbb{F}_p)$ instead of the single $\#\mathbb{F}_p^* = p-1$, by varying the curve E. All these orders have roughly the same size as $p-1$, hence supposedly the same chance of being smooth, using the heuristic principle discussed above.

4.2 Elliptic curves

4.2.1 First step

We first begin with a particular case and define an elliptic curve over the field $\mathbb{R}$ of real numbers. Let us consider an equation

$$E\colon y^2 z = x^3 + axz^2 + bz^3,$$

where $a, b \in \mathbb{R}$ such that $4a^3 + 27b^2 \neq 0$. This define a "curve" $E(\mathbb{R})$ in the projective space

$$\mathbb{P}^2(\mathbb{R}) = \Big\{(x, y, z) \in \mathbb{R}^3 \backslash \{(0, 0, 0)\}\Big\} / \sim,$$

where $\sim$ is the equivalence relation

$$(a, b, c) \sim (a', b', c') \iff \exists\, t \neq 0 \text{ such that } a' = ta,\ b' = tb,\ c' = tc.$$

This "curve" is composed of the class of $(0, 1, 0)$ denoted by $(0 : 1 : 0)$ and of the classes of $(x, y, 1)$ with $y^2 = x^3 + ax + b$ denoted $(x : y : 1)$, so that it can be viewed as the union of a point at infinity O_E and of the curve $\mathcal{C} \subset \mathbb{R}^2$ of equation

$$E' : \ y^2 = x^3 + ax + b,$$

called the Weierstrass equation of $E(\mathbb{R})$.

Now, on $E(\mathbb{R})$ we define an additive law by:

1. $-O_E = O_E$
2. If $P = (x, y) \in \mathcal{C}$, $-P = (x, -y)$ (with the notation $(a, b) = (a : b : 1)$)
3. $O_E + O_E = O_E$
4. If $P \in \mathcal{C}$, $P + O_E = O_E + P = P$
5. If $P = (x, y)$, $Q = (u, v) \in \mathcal{C}$ and $x \neq u$, $P + Q = -S$ where S is the only point of $\mathcal{C} \cap (P, Q)$
6. If $Q = P$, $P + Q = -S$ where S is the only point of $\mathcal{C} \cap \mathcal{T}$ where $\mathcal{T}$ is the tangent to $\mathcal{C}$ in P
7. If $Q = -P$, $P + Q = O_E$.

We can check that these definitions are licit and that, equiped with this law, $E(\mathbb{R})$ is an abelian group with neutral element O_E. In the non trivial cases, explicit formulas are: If $P_1 = (x_1, y_1) \neq O_E$ and $P_2 = (x_2, y_2) \neq O_E$,

- If $x_1 \neq x_2$, $P_3 = (x_3, y_3)$ with $x_3 = \alpha^2 - x_1 - x_2$, $y_3 = -y_1 + \alpha(x_1 - x_3)$ where $\alpha = (y_2 - y_1)/(x_2 - x_1)$.
- If $x_1 = x_2$ and $y_1 = y_2$, $P_3 = P_1 + P_2 = (x_3, y_3)$ with the same formulas where $\alpha = (3x_1^2 + a)/(2y_1)$.

All this extends to the case of a field K of characteristic different from 2 and 3 instead of $\mathbb{R}$. Important cases are $\mathbb{Q}$, $\mathbb{C}$ and $\mathbb{F}_p$. When char $K = 2$ or 3, we can replace equation E by another formula, but this is not an interesting case in what follows.

4.2.2 Elliptic curves over $\mathbb{Z}/N\mathbb{Z}$

Let $N > 0$ be an integer coprime to 6. An "elliptic curve" over $\mathbb{Z}/N\mathbb{Z}$ is an equation

$$E\colon Y^2 Z = X^3 + aXZ^2 + bZ^3, \quad a, b \in \mathbb{Z}/N\mathbb{Z}, \quad 4a^3 + 27b^2 \in (\mathbb{Z}/N\mathbb{Z})^*.$$

This defines a "nonsingular curve" in the projective space $\mathbb{P}^2(\mathbb{Z}/N\mathbb{Z})$:

$$\big\{(x, y, z) \in (\mathbb{Z}/N\mathbb{Z})^3, \gcd(x, y, z, N) = 1\big\} \,/\text{multiplication by } \lambda \in (\mathbb{Z}/N\mathbb{Z})^*.$$

The class of (x, y, z) in $\mathbb{P}^2(\mathbb{Z}/N\mathbb{Z})$ is denoted $(x : y : z)$ as usual. If $p \mid N$, there is a natural map from E to E_p, the curve over $\mathbb{Z}/p\mathbb{Z}$ defined by reducing the equation of E modulo p, i.e. we can reduce points in $E(\mathbb{Z}/N\mathbb{Z})$ to $E_p(\mathbb{Z}/p\mathbb{Z})$. Basic facts about elliptic curves can be found in Silverman [21].

Theorem 4.28. *If N is prime*

1. *$E(\mathbb{Z}/N\mathbb{Z})$ has a natural group structure with neutral element $O_E = (0 : 1 : 0)$. The group law is given by rational mappings, formally the same ones as in the "chord and tangent process" over $\mathbb{R}$ or $\mathbb{Q}$.*
2. *$E(\mathbb{Z}/N\mathbb{Z})$ has at most two cyclic components.*
3. *$(\sqrt{N} - 1)^2 < \#E(\mathbb{Z}/N\mathbb{Z}) < (\sqrt{N} + 1)^2$ (Hasse's bound).*

Now a non-theorem: if N is not prime, $E(\mathbb{Z}/N\mathbb{Z})$ is definitely not a group. But we may still try to add points applying the same defining formulae as if N were prime. The worse obstruction we may encounter is a non-invertible $d \in (\mathbb{Z}/N\mathbb{Z}) \setminus \{0\}$. In this case, $\gcd(d, N)$ is a non-trivial factor of N. In the text, we now assume that all computation depicted in curves over $\mathbb{Z}/N\mathbb{Z}$ (e.g. compute $P + Q$) do succeed. Whenever they do not we obtain a non-trivial factor of N, which is usually our main motivation.

More generally, we define $[m]P$ for $m \in \mathbb{Z}$ and $P \in E(\mathbb{Z}/N\mathbb{Z})$ by

$$[0]P := O_E, \quad [m]P := [m-1]P + P, \text{ for } m > 0$$

and $[m]P := [-m](-P)$ for $m < 0$, where $-P$ is defined as usual. If N is prime, we have genuine associative group law and $[m]P = P + \cdots + P$ with m summands, for $m > 0$; otherwise, the result is undefined if one of the addition fails. The only result we will need about this pseudo group law is that if $p \mid N$ is a prime, $\pi : \mathbb{P}^2(\mathbb{Z}/N\mathbb{Z}) \to \mathbb{P}^2(\mathbb{Z}/p\mathbb{Z})$ is the natural projection, and $P, Q \in E(\mathbb{Z}/N\mathbb{Z})$ such that $P + Q$ is well-defined, then

$$\pi(P + Q) = \pi(P) + \pi(Q), \quad \text{and} \quad \pi([m]P) = [m]\pi(P),$$

where the right-hand side additions use the ordinary group law on $E_p(\mathbb{Z}/p\mathbb{Z})$.

We will subsequently use freely operations in $E(\mathbb{Z}/N\mathbb{Z})$, with the convention that any such operation failing provides a non trivial factor of N and aborts all computations.

4.2.3 Goldwasser-Killian's Elliptic curve primality test

We now adapt Pocklington's theorem 4.22, in the simplest setting, not meant to be practical.

Theorem 4.29. *Let E be an elliptic curve over $\mathbb{Z}/N\mathbb{Z}$, $m = \#E(\mathbb{Z}/N\mathbb{Z})$, $P \in E(\mathbb{Z}/N\mathbb{Z})$, such that*

- *There exists a prime divisor q of m with $q \geq (N^{1/4}+1)^2$,*
- *$[m]P = O_E$, where $[m]P := P + \cdots + P$ with m summands,*
- *$[m/q]P \neq O_E$.*

Then N is prime.

Proof. Assume N composite, and let $p \leq \sqrt{N}$ be the smallest prime divisor of N. On E_p, the order r of P divides m but not m/q, hence $q \mid r$. On the other hand, by Hasse's bound,

$$r \leq \#E(\mathbb{Z}/p\mathbb{Z}) < (\sqrt{p}+1)^2 \leq (N^{1/4}+1)^2,$$

a contradiction. □

If N is prime the above just uses a curve E and $P \in E(\mathbb{Z}/N\mathbb{Z})$ of provably large order (divisible by q). It is natural to choose $m = \#E(\mathbb{Z}/N\mathbb{Z})$ and try random points on the curve:

Proposition 4.30. *Let $N > 3$ be prime, E an elliptic curve over $\mathbb{Z}/N\mathbb{Z}$, $m = \#E(\mathbb{Z}/N\mathbb{Z})$, and q a prime factor of m. Then a random point $P \in E(\mathbb{Z}/N\mathbb{Z})$ satisfies $[m/q]P = O_E$ with probability $1/q$.*

Proof. Since $m < (\sqrt{N}+1)^2$, we have $v_q(m) = 1$. The requested property is true for any abelian group G satisfying $v_q(\#G) = 1$. Indeed, we may write $G = \mathbb{Z}/q\mathbb{Z} \oplus H$ for some abelian group H or order h (coprime to q); the condition $h(a \oplus b) = 0$ is equivalent to $a = 0$, which is true for a fraction $h/\#G = 1/q$ of all elements of G. □

Algorithm 22. Goldwasser-Killian primality test

```
Input: $N > 1$, coprime to 6.
Output: A primality proof for $N$.
1: Pick $a, b \in \mathbb{Z}/N\mathbb{Z}$ at random and let $E : Y^2Z = X^3 + aXZ^2 + bZ^3$.
2: Compute $m = \#E(\mathbb{Z}/N\mathbb{Z})$.
3: Try to factor $m$: if it factors completely leaving a factor $q \geq (N^{1/4} + 1)^2$
   which looks prime (fails a few compositeness tests).
4: Recursively prove the primality of $q$. If it fails, start over at (1).
5: Find $P \in E(\mathbb{Z}/N\mathbb{Z})$ such that $[m/q]P \neq O_E$ and $[m]P = O_E$.   {pick a
   random $x \in \mathbb{Z}/N\mathbb{Z}$ until $x^3 + ax + b$ is a square, then let $y$ be a square
   root and set $P = (x : y : 1)$.}
```

In the above we factor the quadratic polynomial $Y^2 - (x^3 + ax + b)$ as if N were prime: if it fails, N is composite. We can simplify the last step by picking simultaneously E and P: pick randomly $x, y, a \in \mathbb{Z}/N\mathbb{Z}$ and set $b = y^2 - x^3 - ax$.

The big problem with the algorithm is that m is difficult to compute. Goldwasser and Killian use Schoof's algorithm which runs in time $O(\log N)^8$. Even with a lot of improvements since (in particular by Elkies and Atkin), this is still impractical to prove the primality of large integers, say 10000 digits. Atkin's idea is to consider curves with complex multiplication, so that m is known in advance from a simple formula.

4.2.4 Introduction to complex multiplication

An elliptic curve over $\mathbb{C}$ is a torus $\mathbb{C}/\Lambda$, where Λ is a lattice, i.e. a rank 2 submodule. In particular, Λ is not contained in $\mathbb{R}$.

Definition 4.31. An *elliptic function* with period lattice Λ is a meromorphic function f, such that $f(z+w) = f(z)$ for all $w \in \Lambda$. In other words it induces a well-defined function on $\mathbb{C}/\Lambda$ with finitely many poles deleted.

One of the simplest examples is Weierstrass $\wp$-function:

$$\wp(z; \Lambda) := \wp(z) = \frac{1}{z^2} + \sum_{w \in \Lambda \setminus \{0\}} \left(\frac{1}{(z-w)^2} - \frac{1}{w^2} \right).$$

It satisfies the following differential equation:

$$\wp'(z)^2 = 4\wp(z)^3 - g_2\wp(z) - g_3, \quad \text{where } g_2 = 60 \sum_{w \in \Lambda \setminus \{0\}} \frac{1}{w^4}, \quad g_3 = 140 \sum_{w \in \Lambda \setminus \{0\}} \frac{1}{w^6}.$$

(The proof is easy: an elliptic function without poles is constant, because it is entire and bounded. Apply this to the difference $4\wp(z)^3 - g_2\wp(z) - g_3 - \wp'(z)^2$.) Note that g_2 and g_3 are well-defined, in fact

Lemma 4.32. *For any lattice $\Lambda \subset \mathbb{C}$, the series*

$$\sum_{w\in\Lambda\setminus\{0\}} \frac{1}{|w|^s},$$

converges if $s > 2$.

Proof. Prove that $C_k = \#\{w \in \Lambda, |w| \leq k\} = C(\Lambda)k^2 + O(k)$, for some constant $C(\Lambda)$. It follows that $\#\{w \in \Lambda, k \leq |w| < k+1\} = O(k)$. Summing by parts, the sum converges for $s > 2$. □

Proposition 4.33. *We have $g_2^3 - 27g_3^2 \neq 0$, i.e the projective curve $y^2z = 4x^3 - g_2xz^2 - g_3z^3$ is non-singular.*

Remark 4.34. The mapping

$$\begin{array}{rcl} \mathbb{C}/\Lambda & \to & \mathbb{P}^2(\mathbb{C}) \\ z & \mapsto & \begin{cases} (\wp(z) : \wp'(z) : 1) & \text{if } z \notin \Lambda, \\ (0:1:0) & \text{if } z \in \Lambda, \end{cases} \end{array}$$

is a complex isomorphism between the torus $\mathbb{C}/\Lambda$ and the non-singular projective curve $E : y^2z = 4x^3 - g_2xz^2 - g_3z^3$. This endows E with a natural group structure, which coincides with the chord-and-tangent law.

We now consider morphisms between complex elliptic curves, that are holomorphic, $\mathbb{Z}$-linear maps:

Theorem 4.35. *We have the following results.*

1. *The curves $\mathbb{C}/\Lambda$ and $\mathbb{C}/\Lambda'$ are isomorphic if and only if there exists $\alpha \in \mathbb{C}^*$, $\Lambda' = \alpha\Lambda$.*
2. $\mathrm{End}(\mathbb{C}/\Lambda) = \{\alpha \in \mathbb{C} \colon \alpha\Lambda = \Lambda\}$.
3. $\mathrm{End}(\mathbb{C}/\Lambda) = \mathbb{Z}$ *or an order $\mathcal{O}$ in an imaginary quadratic field (i.e. $\mathcal{O} = \mathbb{Z} + w\mathbb{Z}$ with $\mathrm{Im}(w) > 0$ and $w^2 - sw + d = 0$ for some $s, d \in \mathbb{Z}$).*
4. *Let $j(E) = j(\Lambda) = 1728g_2^3/(g_2^3 - 27g_3^2)$. This j-invariant characterizes the isomorphism class of $E = \mathbb{C}/\Lambda$: $j(E) = j(E')$ if and only if $E \cong E'$.*
5. *$j(\alpha\Lambda) = j(\Lambda)$ for any $\alpha \in \mathbb{C}^*$.*

Corollary 4.36. *Note that it is easy to write a Weierstrass equation for a curve with given j-invariant:*

1. *If $j = 0$, take $y^2 = x^3 - 1$.*
2. *If $j = 1728$, take $y^2 = x^3 - x$.*
3. *If $j \neq 0, 1728$, let $c = j/(j - 1728)$ and take $y^2 = x^3 + 3cx + 2c$. The right hand side has discriminant $-2^2 3^3 c^2 (c + 1) \neq 0$.*

The formulas are valid over any base whose characteristic is not 2 *or* 3*.*

Definition 4.37. E has complex multiplication by $\mathcal{O}$ if $\operatorname{End}(E) = \mathcal{O}$ is strictly larger than $\mathbb{Z}$. We say E has CM by $\mathcal{O}$.

Example 4.38. Let us give two examples.

1. The curve $\mathbb{C}/\mathbb{Z}[i]$ has CM by $\mathbb{Z}[i]$. It is isomorphic to $E : y^2 = x^3 - x$, which has extra endomorphism $(x, y) \mapsto (-x, iy)$.
2. The curve $\mathbb{C}/\mathbb{Z}[(1 + \sqrt{-3})/2]$ has CM by $\mathbb{Z}[(1 + \sqrt{-3})/2]$. It is isomorphic to $E : y^2 = x^3 - 1$, which has extra endomorphism $(x, y) \mapsto ((-1 + \sqrt{-3})/2 \cdot x, y)$.

From now on, we insist that a basis (ω_1, ω_2) for Λ be positively oriented: $\operatorname{Im}(\omega_1/\omega_2) > 0$, which is easily achieved by swapping ω_1 and ω_2. Since $j(\alpha\Lambda) = j(\Lambda)$, we may assume that $\Lambda = \langle \tau, 1 \rangle_{\mathbb{Z}}$, with $\operatorname{Im}\tau > 0$. Given a basis (ω_1, ω_2) such that $\tau = \omega_1/\omega_2$ has positive imaginary part, we set $j(\tau) := j(\langle \tau, 1 \rangle_{\mathbb{Z}})$.

Example 4.39. We have $j(i) = 1728$ and $j((1 + \sqrt{-3})/2) = 0$. Moreover if $\langle \tau, 1 \rangle_{\mathbb{Z}}$ is the ring of integers of an imaginary quadratic field $\neq \mathbb{Z}[i]$ and $\neq \mathbb{Z}[(1 + \sqrt{3})/2$, we have $j(\tau) \neq 0$.

Remark 4.40. The reason for the weird normalizing constant 1728 for j is to ensure the following identity

$$j(\tau) = \frac{1}{q} + 744 + \sum_{n \geq 1} c_n q^n, \quad \text{with } q = \exp(2i\pi\tau),$$

where the c_n are positive *integers*.

The fact that $j(\Lambda)$ is a function on lattices, which we compute using any positively oriented basis is equivalent to saying that $j(\tau)$ is invariant under the natural action of $\mathrm{Sl}_2(\mathbb{Z})$:

$$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \cdot \tau = \frac{a\tau + b}{c\tau + d}.$$

Namely $(\omega_1, \omega_2) \to (a\omega_1 + b, c\omega_2 + d)$, $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{Gl}_2(\mathbb{Z})$, gives the general base change and

$$\mathrm{Im}\left(\frac{a\tau + b}{c\tau + d}\right) = \mathrm{im}\left(\frac{(a\tau + b)\overline{(c\tau + d)}}{|c\tau + d|^2}\right) = \frac{(ad - bc)\mathrm{Im}\tau}{|c\tau + d|^2} > 0$$

restricts us to $\mathrm{Sl}_2(\mathbb{Z})$. Hence, we may further impose that τ belongs to the standard fundamental domain for the action of $\mathrm{Sl}_2(\mathbb{Z})$ on Poincaré's half-plane $\mathrm{Im}(z) > 0$:

$$|\mathrm{Re}(z)| \leq \frac{1}{2}, \quad |z| \geq 1.$$

In particular $\mathrm{Im}\tau \geq \sqrt{3}/2$ and $|q| \leq \exp(-\pi\sqrt{3}) \approx 0.0043$. From this, $j(\tau)$ is easy to approximate numerically. In fact, we find

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

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

This is not the fastest way to compute j, but already quite efficient: $n^5 q^n$ tends very quickly to 0.

4.2.5 Some algebraic number theory

A $z \in \mathbb{C}$ is *algebraic* if it is a root of a non-zero polynomial in $\mathbb{Z}[X]$. It is an *algebraic integer* if that polynomial is *monic*: the definition does not depend on the polynomial chosen.

A number field $K \subset \mathbb{C}$ of degree n is a finite extension of $\mathbb{Q}$ of degree n, i.e. a field which is a $\mathbb{Q}$-vector space of finite dimension n. The algebraic integers in K form a ring $\mathbb{Z}_K$. An *integral ideal* is a non-zero ideal in $\mathbb{Z}_K$, a *fractional ideal* is a subset $\mathfrak{A}$ of K such that $d\mathfrak{A}$ is integral for some $d \in K$. A fractional ideal, in particular $\mathbb{Z}_K$, is a free $\mathbb{Z}$-modules of rank n.

Remark 4.41. We define number fields as embedded in $\mathbb{C}$. This is not a necessity, and is in fact the wrong point of view. We could view K as an abstract field embedding $\mathbb{Q} \subset K$ or, more concretely but less adequately, as a quotient ring $K = \mathbb{Q}[X]/(T)$, for some irreducible $T \in \mathbb{Q}[X]$ of degree n (this follows from the primitive element theorem, and is not suitable to define extensions over more general bases than $\mathbb{Q}$, e.g. $\mathbb{F}_p(t)$). To each complex root α_i of T corresponds an embedding $K \to \mathbb{C}$, sending the class of X to α_i, and each complex embedding is of this form. In this way, an abstract number field of degree n comes equipped with n canonical embeddings into the complex numbers, or any algebraically closed field. There is no reason to favor any of them.

For instance, the field $\mathbb{Q}[X]/(X^3 - 2)$ has three different embeddings, $\mathbb{Q}(2^{1/3})$, $\mathbb{Q}(2^{1/3}j)$ and $\mathbb{Q}(2^{1/3}j^2)$, which are isomorphic as fields, but definitely not identical. For one thing, the first one is a subset of $\mathbb{R}$.

There is a natural multiplication on the set of integral ideals: $\mathfrak{A}\mathfrak{B}$ is the smallest integral ideal containing all ab, $a \in \mathfrak{A}$, $b \in \mathfrak{B}$. This extends in a natural way to fractional ideals.

Theorem 4.42. *We have:*

1. *$\mathbb{Z}_K$ is a Dedekind ring: the fractional ideals of K form an abelian group under multiplication, with neutral element $\mathbb{Z}_K$. Any fractional ideal can be written uniquely as a product of maximal ideals: $\mathfrak{A} = \prod \mathfrak{p}^{e_\mathfrak{p}}$, where all $e_\mathfrak{p}$ but finitely many are 0. $\mathfrak{A}$ is integral, if and only if $e_\mathfrak{p} \geq 0$ for all $\mathfrak{p}$. If $\mathfrak{p} \subset \mathbb{Z}_K$ is a maximal ideal, then $\mathbb{Z}_K/\mathfrak{p}$ is a finite field.*

2. *If $\mathfrak{A}$ is an integral ideal, then the norm of $\mathfrak{A}$, $N\mathfrak{A} := \mathbb{Z}_K/\mathfrak{A}$ is finite, this is a multiplicative map: $N\mathfrak{A}\mathfrak{B} = N\mathfrak{A} \cdot N\mathfrak{B}$, which extends to fractional ideals by multiplicativity.*

3. *Let I_K be the group of fractional ideals, and P_K the subgroup of* principal *fractional ideals. The quotient group I_K/P_K is a finite abelian group $\mathrm{Cl}(K)$, the class group of K.*

4. *There exists a finite Galois extension H_K/K, the Hilbert class field of K, whose Galois group is $\mathrm{Cl}(K)$. A maximal ideal $\mathfrak{p} \subset \mathbb{Z}_K$ splits completely in H_K if and only if it is principal.*

(The ideal $\mathfrak{p}$ splits completely in L/K if $\mathfrak{p}\mathbb{Z}_L$ is a product of $[L : K]$ distinct maximal ideals in $\mathbb{Z}_L$; they all satisfy $\mathbb{Z}_L/\mathfrak{P} = \mathbb{Z}_K/\mathfrak{p}$.)

We shall see in the next lecture how to compute class groups of imaginary quadratic fields. From this, the j-invariant will enable us to compute Hilbert class fields. There are beautiful and important algorithms for the general case of arbitrary number fields, and generalizations of the Hilbert class field, just beyond the scope of our lectures. Class fields provide a complete solution to the inverse Galois problem for abelian groups. In short, they describe all abelian extensions of a given number field K, in terms of arithmetic data depending on K only. For an introduction to this so-called Class Field Theory in a context very close to ours, see Cox [8], then Cohen [6, 7] for a computational approach.

Let us consider a few examples:

1. $K = \mathbb{Q}$, one finds $\mathbb{Z}_K = \mathbb{Z}$, $\mathrm{Cl}(K) = \{1\}$ ($\mathbb{Z}$ is principal, in fact Euclidean), hence $H_K = K$.

2. $K = \mathbb{Q}(i)$, one finds $\mathbb{Z}_K = \mathbb{Z}[i]$, $\mathrm{Cl}(K) = \{1\}$ ($\mathbb{Z}[i]$ is principal, again Euclidean), hence $H_K = K$.

3. $K = \mathbb{Q}(\sqrt{-6})$, one finds $\mathbb{Z}_K = \mathbb{Z}[\sqrt{-6}]$, $\mathrm{Cl}(K) \simeq \mathbb{Z}/2\mathbb{Z}$, generated by the maximal ideal $\mathfrak{p}$ generated by 2 and $\sqrt{-6}$, which is *not* principal; in this case $\mathbb{Z}_K/\mathfrak{p} = \mathbb{F}_2$. The Hilbert class field H_K is
$$K(\sqrt{-3}) = K(\sqrt{2}) = \mathbb{Q}(\sqrt{2} + \sqrt{-3}).$$

Remark 4.43. The notation $\sqrt{d}$ for $d < 0$ denotes the complex number $i\sqrt{d} \in \mathbb{C}$. Had we taken the better point of view that $K = \mathbb{Q}(\sqrt{d})$ is really $\mathbb{Q}[X]/(X^2 - d)$, for some non-square d, we could just say it is the class of X in K.

Exercice 4.44. Let d be a squarefree integer, $K = \mathbb{Q}(\sqrt{d})$. Prove that $\mathbb{Z}_K = \mathbb{Z}[\omega]$, where
$$\omega = \begin{cases} \sqrt{d} & \text{if } d \equiv 2,3 \pmod 4, \\ \dfrac{1+\sqrt{d}}{2} & \text{if } d \equiv 1 \pmod 4. \end{cases}$$

(When is $(u + \sqrt{d})/2$ an algebraic integer, if $u, v \in \mathbb{Q}$?)

The *discriminant* of a quadratic field $\mathbb{Q}(\sqrt{d})$, d squarefree, is $D = 4d$ if $d \equiv 2,3 \pmod 4$, and d otherwise. Then $\mathbb{Z}_K = \mathbb{Z}[\frac{D+\sqrt{D}}{2}]$ in all cases.

4.2.6 Class groups of imaginary quadratic fields

Let $K = \mathbb{Q}(\sqrt{D})$ be the imaginary quadratic field of discriminant $D < 0$. The most efficient representation for handle ideal classes in $\mathbb{Z}_K$ uses an isomorphism with classes of integral binary quadratic forms modulo the action by $\mathrm{Sl}_2(\mathbb{Z})$ given by change of variables. We will stick to ideals, which entails minor inefficiencies. See [6, Chap.5] for details.

The proofs of the following lemmas are not difficult and left as exercises:

Lemma 4.45. *If $\alpha \in K \subset \mathbb{C}$, then $N(\alpha) = |\alpha|^2$.*

Lemma 4.46. *Let K be a quadratic field of discriminant D. All integral ideals in $\mathbb{Z}_K$ are of the form $\mathfrak{A} = \delta\big(a\mathbb{Z} + \frac{-b+\sqrt{D}}{2}\mathbb{Z}\big)$ for some $a, \delta \in \mathbb{Z}_{>0}$, $b \in \mathbb{Z}$, such that $-a < b \leq a$ and $b^2 \equiv D \pmod{4a}$. Conversely, these $\mathbb{Z}$-modules are* distinct *ideals (= $\mathbb{Z}_K$-modules). Further, $\mathfrak{A} \cap \mathbb{Z} = \delta a\mathbb{Z}$ and $N\mathfrak{A} = a\delta^2$.*

Proof. Tedious but simple computations. (Write $\mathbb{Z}_K = \left\langle 1, \frac{D+\sqrt{D}}{2} \right\rangle_{\mathbb{Z}}$ and $\mathfrak{A}$ as a submodule given by an HNF basis.) Note that the lemma holds for real and imaginary fields. □

Definition 4.47. An integral ideal is *primitive* if it is not of the form $\delta\mathfrak{A}$ with $\mathfrak{A}$ integral and $\delta > 1$. We represent the primitive ideal $\big(a\mathbb{Z} + \frac{-b+\sqrt{D}}{2}\mathbb{Z}\big)$ by the triple $(a, b, c) \in \mathbb{Z}^3$, with $a > 0$, $-a < b \leq a$ and $c := (b^2 - D)/(4a) > 0$. Since c can be deduced from a, b and D, we may omit it in the notation, as in $(a, b, *)$.

The condition $-a < b \leq a$ only says that $b = \mathrm{MOD}\,(b, 2a)$, and corresponds to the obvious fact that $\langle a, \tau\rangle_{\mathbb{Z}} = \langle a, \tau - qa\rangle_{\mathbb{Z}}$ for any $q \in \mathbb{Z}$. Since we are mostly interested in ideal classes, it does not hurt to assume that ideals are primitive. Note also that if $\tau = aX - \frac{-b+\sqrt{D}}{2}Y$ lies in the ideal (a, b, c), i.e. if $X, Y \in \mathbb{Z}$, then

$$N(\tau) = |\tau|^2 = a(aX^2 + bXY + cY^2) \tag{4.1}$$

(Here come the binary quadratic forms.)

Lemma 4.48. *With the previous notations, let*

$$\mathfrak{A} = (a, b, c), \quad \mathfrak{B} = (c, \mathrm{MOD}\,(-b, 2c)\,, *), \quad \tau = \frac{b + \sqrt{D}}{2a} \in K.$$

Then $\mathfrak{A}$ and $\mathfrak{B}$ are in the same ideal class. In fact, $\mathfrak{B} = (\tau)\mathfrak{A}$.

Corollary 4.49. *Each ideal class in $\mathbb{Z}_K$ contains a unique ideal of the form $\mathfrak{A} = (a, b, c)$ such that*

$$|b| \leq a \leq c,$$

and $b \geq 0$ if one inequality is an equality. Such an ideal is called reduced.

Proof. Existence is easy: pick a representative in the ideal class such that $a \in \mathbb{Z}_{>0}$ is minimal. Then $a \leq c$ by minimality and the previous lemma. The other conditions are easy.

We now prove unicity. Using (4.1), we claim that

$$a^2 = \min_{x \in \mathfrak{A}\setminus\{0\}} |x|^2\,, \quad ac = \min_{x \in \mathfrak{A}\setminus\mathbb{Z}} |x|^2\,.$$

Let us prove the first one: a^2 is obviously attained ($X = 1$, $Y = 0$); using the defining equalities $|b| \leq a \leq c$, we have

$$aX^2 + bXY + cY^2 \geq a(X^2 - |XY| + Y^2) \geq a,$$

since $X, Y \in \mathbb{Z}$ implies

$$X^2 - |XY| + Y^2 = (X - |Y|/2)^2 + \frac{3}{4}Y^2 \geq 1, \quad (X, Y) \in \mathbb{Z}^2 \setminus \{(0,0)\}.$$

The second equality follows analogously, by investigating the consequences of $Y \neq 0$.

Assume now that $\mathfrak{A} = (a, b, c)$ and $\mathfrak{A}' = (a', b', c')$ are reduced and in the same ideal class: $\mathfrak{A} = \alpha\mathfrak{A}'$ for some $\alpha \in K^*$. This implies $N\mathfrak{A} = N(\alpha)N\mathfrak{A}'$ hence $a = |\alpha|^2 a'$ by the multiplicativity of norms and Lemma 4.45. Then

$$\left\{|x|^2 : x \in \mathfrak{A}' \setminus \{0\}\right\} = |\alpha|^2 \left\{|x'|^2 : x' \in \mathfrak{A}' \setminus \{0\}\right\},$$

hence they have the same minimum and $a^2 = |\alpha|^2 a'^2$. It follows that $a = a'$ and $|\alpha| = 1$. Then $\mathfrak{A} \cap \mathbb{Z} = \mathfrak{A}' \cap \mathbb{Z} = a\mathbb{Z}$ and

$$\left\{|x|^2 : x \in \mathfrak{A}' \setminus \mathbb{Z}\right\} = \left\{|x'|^2 : x' \in \mathfrak{A}' \setminus \mathbb{Z}\right\},$$

from which we obtain $ac = a'c'$, hence $c = c'$. Since $b^2 - 4ac = b'^2 - 4a'c'$, it follows that $b = \pm b'$. By the convention on the sign of b, we can assume $0 < |b| < a < c$, otherwise $b, b' \geq 0$ and we are done.

In these conditions, the only solutions of $aX^2 + bXY + cY^2 = a$ are $(\pm 1, 0)$ (refine the computations above). Since $a \in \mathfrak{A}$, there exists $u = Xa' - Y\frac{-b'+\sqrt{D}}{2} \in \mathfrak{A}'$, $X, Y \in \mathbb{Z}$, such that $a = \alpha u$, hence $|u|^2 = a^2$, which now implies that $(X, Y) = (\pm 1, 0)$. Finally $\alpha = \pm 1$, and $\mathfrak{A} = \mathfrak{A}'$. □

Note that $\mathbb{Z}_K$ is always reduced and is represented either by $(1, 0, -D/4)$ or by $(1, 1, (1 - D)/4)$ depending on D mod 4. More importantly, the triples representing reduced ideals are easily bounded: in fact, let $\Delta = |D| = -D$, the definitions $|b| \leq a \leq c$ and $D = b^2 - 4ac$ imply

$$-D = 4ac - b^2 \geq 4a^2 - a^2 = 3a^2,$$

hence $a \leq \sqrt{-D/3}$. This yields a simple algorithm to enumerate $\mathrm{Cl}(K)$:

Algorithm 23. Class group of imaginary quadratic fields

Input: $D < 0$, discriminant of a quadratic fields K.

Output: A set of reduced ideal representatives of $\mathrm{Cl}(K)$.

```
for $a = 1, \dots, \sqrt{-D/3}$ do
    for $b = 0, \dots, a$ such that $b \equiv D \pmod 2$ do
        Let $c = (b^2 - D)/4a$.
        if $c \in \mathbb{Z}$ and $a \leq c$ then
            Print $(a, b, c)$.
            If $b \neq a \neq c$ and $b \neq 0$, print $(a, -b, c)$.
```

Theorem 4.50. *This algorithm prints all reduced ideals in $\mathbb{Z}_K$ in time $\widetilde{O}(D)$.*

Proof. The two outer loops have length $O(\sqrt{\Delta})$. □

One proves that $\#\mathrm{Cl}(D) \ll \sqrt{\Delta} \log \Delta$, hence the algorithm above is certainly not optimal. In fact the Brauer-Siegel theorem states that $\log \#\mathrm{Cl}(D) \sim \log(\sqrt{\Delta})$ as D tends to infinity, so the size of the output is indeed roughly $\sqrt{\Delta}$. It is possible to enumerate $\mathrm{Cl}(D)$ in time essentially linear in the size of the output by describing first $\mathrm{Cl}(D)$ in terms of generators and relations, see §**??**. Note that the error terms in Brauer-Sieger are *ineffective* unless we assume a Riemann Hypothesis.

Algorithm 24. Reduction of ideals in imaginary quadratic fields

Input: A primitive integral ideal $\mathfrak{A} = (a, b, c)$ in $\mathbb{Z}_K$.

Output: The reduced representative $\mathfrak{B}$ of the ideal class of $\mathfrak{A}$, and a $\tau \in K^*$ such that $\mathfrak{B} = \tau\mathfrak{A}$.

```
Set $\tau \leftarrow 1$.
while $a > c$ or $(b < 0$ and $a = c)$ do
    Let $a' = c$, $b' = \mathrm{MOD}\,(b, 2a')$, $c' = (b'^2 - D)/(4a')$.
    Set $(a, b, c) \leftarrow (a', b', c')$.
    Set $\tau \leftarrow \tau \times \frac{b+\sqrt{D}}{2a}$.
```

Proof. Since we assume $-a < b \leq a$, it is impossible that $b = -a$ and the output is correct by repeated application of Lemma 4.48. The only thing to prove is that the algorithm stops. If $a = c$ and $b < 0$ at the beginning of the loop then (a, b, a) is replaced by $(a, -b, a)$ and we exit the loop in the next iteration. Otherwise, $a \in \mathbb{Z}_{>0}$ decreases strictly from one loop to the next, and we are done. With a little care [6, Prop 5.4.3], one proves $a' \leq a/2$ except in the very last iteration, so there are $O(\log a)$ iterations. □

Algorithm 25. Solving $U^2 - DV^2 = 4N$

```
Input: $D < 0$ discriminant of a quadratic field; $N$ an odd prime coprime to $D$.
Output: Return a solution $(U, V)$ of $U^2 - DV^2 = 4N$, or False if none exist.
1: Factor the quadratic polynomial $X^2 - D$ in $\mathbb{Z}/N\mathbb{Z}$, as if $N$ were prime.
2: If the polynomial is irreducible, return False.
3: Let $0 < b < N$ be a square root of $D$ modulo $N$. If $b$ and $D$ have different parities, set $b \leftarrow N - b$.
       {Now, $b^2 \equiv D \pmod{4N}$.}
4: Let $c = \frac{b^2-D}{4N} \in \mathbb{Z}$. Using the previous reduction algorithm with input $\mathfrak{A} = (N, b, c)$, find a reduced representative $\mathfrak{B} = (a, *, *)$ of $\mathfrak{A}$, and $\tau \in K$ such that $\mathfrak{B} = \tau\mathfrak{A}$.
5: If $a \neq 1$, return False.
6: Write $1/\tau = \frac{U+V\sqrt{D}}{2}$ and return $(U, V)$.
```

Proof. The equation means $N = \alpha\overline{\alpha}$, where $\alpha = (U+V\sqrt{D})/2$ is an algebraic integer, since it is a root of the monic $X^2 - UX + N \in \mathbb{Z}[X]$. The integral ideal (α) has norm N, which is prime, hence it must be primitive and can be represented as $(N, b, *)$ where $b^2 \equiv D \pmod{4N}$. Provided this is possible, the equation has a solution if and only if this ideal is principal, i.e. $\mathfrak{B} = \mathbb{Z}_K$, which is the same as $a = 1$. □

The proof shows that, if $\alpha = (U+V\sqrt{D})/2$ is the solution returned for the equation $\text{Norm}\alpha = N$, the others are of the form $\zeta\alpha$ or $\zeta\bar{\alpha}$, where $\zeta \in \mathbb{Z}_K^*$. Hence there are either 0 or exactly $2w(D)$ solutions, where $w(D) = \#\mathbb{Z}_K^*$ is the number of units in $\mathbb{Z}_K$, i.e. $w(D) = 2$ for $D < -4$, $w(-4) = 4$, $w(-3) = 6$.

4.2.7 Atkin's idea and ECPP

Let $K = \mathbb{Q}(\sqrt{D})$ be an *imaginary* quadratic field of discriminant $D < 0$. If $\mathfrak{A} \subset \mathbb{Z}_K \subset \mathbb{C}$ is an integral ideal, $\mathbb{C}/\mathfrak{A}$ is an elliptic curve with CM by $\mathbb{Z}_K$.

Theorem 4.51. *We have:*

1. *The Weierstrass equation E of $\mathbb{C}/\mathfrak{A}$ is defined over H_K, in particular $j(E) = j(\mathfrak{A}) \in H_K$. Note that $j(\mathfrak{A})$ only depends on the class of $\mathfrak{A}$ in $\mathrm{Cl}(K)$. In fact, if $\mathfrak{A}$ runs through the classes of $\mathrm{Cl}(K)$, the $j(\mathfrak{A})$ define the $\mathbb{C}$-isomorphism classes of elliptic curves with CM by $\mathbb{Z}_K$.*

2. *Let*
$$\Phi(X) = \prod_{\mathfrak{A}\in \mathrm{Cl}(K)} \big(X - j(\mathfrak{A})\big).$$
Then $\Phi(X) \in \mathbb{Z}[X]$ is irreducible and any root of Φ generates H_K/K.

3. *A prime N splits completely in $H_K/\mathbb{Q}$ if and only if the equation $U^2 - DV^2 = 4N$ has a solution in integers U, V. In this case, maximal ideals $\mathfrak{p}$ above N in H_K satisfy $\mathbb{Z}_K/\mathfrak{p} = \mathbb{Z}/N\mathbb{Z}$ and are principal.*

Exercice 4.52. Using the algorithm in the previous lecture, show that $K = \mathbb{Q}(\sqrt{-163})$ has trivial class group. Then explain why $\exp(\pi\sqrt{163})$ is very close to an integer.

For a root $j \in H_K$ of the modular polynomial $\Phi(X)$, let $E(j)$ the complex curve of Corollary 4.36, which is defined over H_K. In the situation of (3), N splits completely and we can intuitively reduce the equation E modulo $\mathfrak{p}$, to obtain $\bar{E}$ over $\mathbb{Z}/N\mathbb{Z}$. Unfortunately, even though j is integral, the equation $E(j)$ does not have integral coefficients, and denominators are a nuisance. The proper way to proceed is as follows: $\Phi(X)$ splits into distinct linear factors modulo N and its roots $\bar{j}$ are the reductions of the j invariants of the $\mathbb{C}$-isomorphism classes of curves with CM by $\mathbb{Z}_K$. The whole point of the construction, besides surprisingly producing explicit curves $\bar{E} = E(\bar{j})$ over $\mathbb{Z}/N\mathbb{Z}$ from complex analytic data, is that the $\#\bar{E}(\mathbb{Z}/N\mathbb{Z})$ are known:

Theorem 4.53. *We have:*

1. *In the above situation, $\#\bar{E}(\mathbb{Z}/N\mathbb{Z}) = N + 1 - U$ for some solution (U, V) of $U^2 - DV^2 = 4N$. Note that Hasse's bound $|U| < 2\sqrt{N}$ is obvious in this case.*

2. *Conversely, there are $w(D)$ such solutions U, where $w(D) = \#\mathbb{Z}_K^*$, and each give rise to the cardinality of a curve.*

Remark 4.54. More precisely let $g \in \mathbb{Z}/N\mathbb{Z}$ such that $g^{(N-1)/p} \not\equiv 1$ for each prime $p \mid w(D)$ ($w(D) \mid N - 1$ since N splits in $\mathbb{Z}_K$). Then we have:

1. If $D = -4$ ($j = 1728$) the curves are $y^2 = x^3 - x$ and its quadratic twists i.e. the curves $y^2 = x^3 - g^k x$ where $1 \leq k \leq 3$.

2. If $D = -3$ ($j = 0$) the curves are $y^2 = x^3 - 1$ and its quadratic twists i.e. the curves $y^2 = x^3 - g^k$ where $1 \leq k \leq 5$.

3. Otherwise ($j \neq 0,\ 1728$) the curves are $y^2 = x^3 - 3cx + 2c$ (with $c = j/(j - 1728)$) and its quadratic twist $y^2 = x^3 - 3cg^2 + 2cg^3$.

We can now formulate roughly the main ideas in Atkin's algorithm. Its complexity is not rigorously analyzed, but optimized variants are the fastest known methods in practice.

Algorithm 26. ECPP primality test

```
Input: $N$ an integer
Output: True if $N$ is prime, False otherwise.
for $D = -3, -4, -7, \dots$ do   {Loop over imaginary quadratic fields $K = \mathbb{Q}(\sqrt{D})$}
    if $D$ is a discriminant and $4N = U^2 - DV^2$ has integer solutions then
        Compute $m_U = N + 1 - U$ for all solutions $(U, V)$. If one of these is
            completely factored up to a large probable prime $q \geq (N^{1/4} + 1)^2$.
Compute representatives $\mathfrak{A}$ for the elements of $\mathrm{Cl}(K)$.
Compute floating point approximations of the $j(\mathfrak{A})$, then of the polynomial $\Phi(X)$ and round its coefficients to the nearest integer.
    {If $N$ is prime, then $\Phi$ splits in $\mathbb{Z}/N\mathbb{Z}$}
Compute a root $\bar{j}$ of $\Phi$ in $\mathbb{Z}/N\mathbb{Z}$, and write a Weierstrass equation for $\bar{E}$ with $j$-invariant $\bar{j}$.
Test a few random points $P$ until we believe that $\#\bar{E}(\mathbb{Z}/N\mathbb{Z}) = m_U$, i.e. we always have $[m_U]P = O_E$. If for even a single $P$, the test fails, replace $E$ by its quadratic twist. (If $D = -3, -4$, consider all quadratic twists until we find one whose cardinality is probably $m_U$.)
Use the Goldwasser-Killian test to prove that $N$ is prime, assuming that $q$ is: pick a random point $P$ and check that $[m_U/q]P = (x : y : z)$, with $\gcd(z, N) = 1$.
Then recursively prove the primality of $q$.
```

The algorithm is not completely formalized and it is clear there are many places where it can be improved. For instance,

- One should consider the D by increasing class numbers. Also, better invariants than j are known (smaller polynomials than Φ) to compute H_K/K.

- With some more theoretical effort, one can avoid almost all the guesswork when matching m_U to a specific curve (not up to quadratic twist as shown above).

- A heuristic analysis shows that the most time consuming part is the solving of $b^2 \equiv D \pmod{N}$, so we can restrict the D to be products of small primes p_i and precompute their square roots. Then the square root of D can be recovered by a multiplication. This is the main idea in the FastECPP algorithm, which is conjectured to run in randomized time $\widetilde{O}(\log N)^4$, where the original algorithm was conjectured to run in randomized times $\widetilde{O}(\log N)^5$. Current records for primality proofs of general integers use FastECPP and lie around 20000 decimal digits (june 2006).

As in §4.1.4, ECPP produces a primality certificate for N, of the form $C(N)$, which consists in

- the integer N,
- a curve E over $\mathbb{Z}/N\mathbb{Z}$ and a point $P \in E(\mathbb{Z}/N\mathbb{Z})$,
- an integer m divisible by $q \geq (N^{1/4}+1)^2$, such that $[m]P = O_E$ and $[m/q]P \neq O_E$,
- a certificate $C(q)$ for q.

4.2.8 Factoring with elliptic curves

This is straightforward generalization of the $p-1$ method from §4.1.6.

Algorithm 27. Lenstra's ECM factorization algorithm

Input: N an integer, B a smoothness bound.

Output: A non-trivial factor of N or `Fail`.

```
Using Eratosthenes's sieve, compute all primes $p \leq B$.
Pick a random curve $E$ over $\mathbb{Z}/N\mathbb{Z}$ and $P \in E(\mathbb{Z}/N\mathbb{Z})$. Let $Q = P$.
for $p \leq B$ do    {compute $Q = [\mathrm{lcm}(2, \ldots, B)]P$}
    Let $k$ be maximal such that $p^k \leq B$.
    Set $Q := [p^k]Q$. If during the computation we have a problem, it
    is because we have found a non-zero and non invertible element $d$ of
    $\mathbb{Z}/N\mathbb{Z}$ (see formulas) and we are happy!
Return Fail.
```

Just as for the $p-1$ method, we can introduce a B_2-phase to cater for $\#E(\mathbb{F}_p)$ which are B_1-powersmooth up to a single larger prime $\leq B_2$.

Conjecture 4.55. With suitable parameters, ECM runs in randomized time $L_{1/2}(p)^{1/\sqrt{2}+o(1)}$, where p is the smallest prime divisor of N. Since $p \leq \sqrt{N}$, this is $L_{1/2}(N)^{1+o(1)}$.

4.3 Sieving methods

4.3.1 The basic idea

We want to find $x, y \in \mathbb{Z}$ such that

$$x^2 \equiv y^2 \pmod N, \quad x \not\equiv \pm y \pmod N.$$

Since this is impossible if $(\mathbb{Z}/N\mathbb{Z})^*$ is cyclic (unless we stumble directly on x, y not belonging to $(\mathbb{Z}/N\mathbb{Z})^*$, which is highly implausible but would indeed yield factors), we must check that N is an not a prime power before embarking on this course of action. We may as well require that it is not a pure power, of the form $N = q^k$. Note that $k = O(\log N)$ and an approximation to $N^{1/k}$ is easily computed for each given k using Newton's method (or approximating $\exp(\frac{1}{k}\log N)$), so this is not a costly pre-condition :

Algorithm 28. Generic sieving factorization algorithm

```
Input: An odd integer $N$, not a pure power.
Output: A non-trivial factor of $N$ or Fail.
1: Choose a factorbase $\mathcal{B}$, e.g. $\{-1\} \cup \{p \le B\}$
2: Produce many congruences of the form
       $x_j^2 \equiv \prod_{i\in\mathcal{B}} i^{e_{i,j}} \pmod N, \quad j \le J,$
   where $e_{i,j} \in \mathbb{Z}$, but may as well be taken in $\{0, 1\}$.
3: If $v = (v_j)$ is a non-zero vector in the kernel of $(e_{i,j})$, viewed as a matrix
   over $\mathbb{F}_2$, then $\sum_j e_{i,j}v_j \equiv 0 \pmod 2$ for all $i \in \mathcal{B}$ and
       $\left(\prod_j x_j^{v_j}\right)^2 \equiv \left(\prod_{i\in\mathcal{B}} i^{\frac{1}{2}\sum_j e_{i,j}v_j}\right)^2 \pmod N,$
   which is of the required form $x^2 \equiv y^2 \pmod N$.
4: If $x \equiv \pm y \pmod N$, return Fail.
```

Of course, in practice, one never returns `Fail`, but computes further kernel vectors. The number of square roots of 1 in $\mathbb{Z}/N\mathbb{Z}$ is 2^ω, where ω is the number of distinct prime divisors of N. So two random (x, y) such that $x^2 \equiv y^2 \pmod N$ yield a trivial factor with probability $2^{1-\omega} \le 1/2$ provided $\omega \ge 2$.

The whole difficulty is now to find relations, and then to choose suitably the factorbase depending of our relation-finding algorithm. Let

$$B := \left\lfloor \sqrt{N} \right\rfloor .$$

The simplest idea (Dixon's random squares) is to pick $t > B$ at random. If $\mathrm{MOD}\,(t^2, N)$ factors on $\mathcal{B}$, we have found a relation. Of course, we want $t \approx B$ to that $\mathrm{MOD}\,(t^2, N)$ be of the order of $\sqrt{N}$ and no larger. This method provably runs in expected time $L_{1/2}(N)^{O(1)}$.

For instance, we may try $t = B + a$, for $a = 1, 2, \ldots$. Note that if a is not too large, then

$$\mathrm{MOD}\left(t^2, N\right) = (B + a)^2 - N,$$

which is the basic idea behind the quadratic sieve.

4.3.2 The quadratic sieve

It is quite costly to check directly that a given integer is B-smooth: about $B/\log B$ divisions. Sieves are slower, but they can check a whole range of numbers simultaneously, at an essentially constant cost per number. Just like Eratosthenes's sieve produces many more primes than a single primality proof. The main problem with the trial division approach is that almost all numbers tested are not smooth, so we have a huge amount of wasted work.

Let $Q(X) = (X + B)^2 - N$, with $B = \left\lfloor \sqrt{N} \right\rfloor$. Since this is a polynomial in $\mathbb{Z}[X]$, if $m \mid Q(a)$, then $m \mid Q(a + \lambda m)$ for all $\lambda \in \mathbb{Z}$.

Algorithm 29. Quadratic sieve

Input: An integer N, a factorbase $\mathcal{B}$, a sieving bound $M > 1$. We assume that all $p \in \mathcal{B}$ satisfy $\left(\frac{N}{p}\right) = 1$.

Output: A set of $a \leq M$ such that $Q(a)$ is $\mathcal{B}$-smooth: $p \mid Q(a)$ implies $p \in \mathcal{B}$.

```
Build an array $A[a] = Q(a)$, for $1 \leq a \leq M$.
If $2 \in \mathcal{B}$, replace $A[a]$ by its largest odd factor for all $a \leq M$.
for $p \in \mathcal{B}$ odd prime do
    Find the largest $k$ such that $p^k \leq Q(M)$.
    Find $a_k, b_k$ such that $Q(a_k), Q(b_k) \equiv 0 \pmod{p^k}$.   {Two solutions mod $p$ then Hensel lift.}
    for $i = k, k-1, \dots, 1$ do
        If $i < k$, set $(a_i, b_i) \leftarrow (a_{i+1}, b_{i+1}) \bmod p^i$.
        for $-a_i/p^i \leq \lambda \leq (M - a_i)/p^i$, $p \nmid \lambda$ do
            Divide $A[a_i + \lambda p^i]$ by $p^i$ in place.   {exact division}
            Divide $A[b_i + \lambda p^i]$ by $p^i$ in place.   {exact division}
Return all the $a$ such that $A[a] = 1$.
```

Proof. The reason for the condition $\left(\frac{N}{p}\right)$ is that p cannot divide $Q(a)$ unless N is a square modulo p. Of course, if $p \mid N$, we have factored N. So the assumption is harmless. Since we ensure $p \neq 2$, it follows that the equation $x^2 \equiv N \pmod{p^k}$ has exactly 2 solutions for all k. The result is correct since we divide the $A[a]$ by primes belonging to $\mathcal{B}$. Note that we may miss some smooth numbers this way, but this agrees with the specifications. □

Provided M is large enough, the sieve proper dominates the running time. In an actual implementation, two basic improvements are useful:

- Initialize $A[a]$ by a rough approximation to $\log Q(a)$, then subtract approximations to $i \log p$ instead of dividing by p^i: subtractions are cheaper than divisions. Of course, the $\log p$ are precomputed.

- Do not sieve by small primes, which cost a lot (there are lots of them) and do not decrease much the size of $A[a]$. Then we may as well not sieve by prime powers, since most integers will not be divisible by too many squares of a not-so-small primes.

In the end, we check directly by trial division the a such that $A[a]$ is not too large. Due to the above two approximations, we may miss quite a few smooth numbers, which seems wasteful. But we still expect a large number of the corresponding $Q(a)$ to be smooth. Since the sieve is now much faster, proper tuning results in a net gain: finding twice fewer relations in each sieving range is not a major problem if we find them three times faster!

4.3.3 The Multiple Polynomials Quadratic Sieve (MPQS)

Our polynomial Q is nice but it stands all alone, and the $Q(a)$ increase relatively fast with a. We now replace the polynomial $(X+B)^2 - N$ with a more general polynomial $Q(X) = AX^2+2BX+C$, with reduced discriminant $B^2 - AC = N$, hence $AQ(X) = (AX+B)^2 - N$.

What are suitable parameters ? $Q(X)$ is minimal at $-B/A$, with value $-N/A$, which is fine compared to our old $(X+B)^2 - N$ if A is not much smaller than $\sqrt{N}$. We can get a symmetric range of small values: for all $x \in [-B/A - M, -B/A + M]$, we have

$$-N = AQ(-B/A) \leq AQ(x) \leq AQ(-B/A + M) = (AM)^2 - N,$$

so if $A \leq \sqrt{2N}/M$, we have $|Q(x)| \leq N/A \approx M\sqrt{N/2}$.

```
Algorithm 30. Recipe to find $Q$ for the quadratic sieve
Input: An odd integer $N$, a sieving bound $M > 1$.
Output:
1: Find $A$ odd prime such that $A \approx \sqrt{2N}/M$ and $\left(\frac{N}{A}\right) = 1$.
2: Find $B$, $B^2 \equiv N \pmod{A}$ and let $C := (B^2 - N)/A$
3: Return $Q = AX^2 + BX + C$.
```

We take A prime and $\left(\frac{N}{A}\right) = 1$ so that the congruence be solvable (factor a quadratic polynomial over a finite field). Such A are found by trial and error, trying consecutive integers larger than $\sqrt{2N}/M$ until they fail a few compositeness tests and the Legendre symbol has the right value. Now when the $Q(a)$ become large, we can find a new polynomial Q with different arithmetic properties, and still relatively small values!

We must also compute the roots of Q modulo primes in the factorbase. Provided $p \nmid A$, the roots of $Q(a) \equiv 0 \pmod{p}$ are the $(-B+a_1)A^{-1} \pmod{p}$, $(-B+b_1)A^{-1} \pmod{p}$, with a_1, b_1 the square roots of $N \pmod{p}$ as before. If $p \mid A$ there is a single root $-BC^{-1}$ modulo p. Note that it is more difficult to sieve modulo prime powers when $p \mid A$: another reason to avoid it.

4.3.4 The Self Initializing MPQS, Large Prime variations

It is unfortunately rather costly to change Q, so we cannot change it as often as we please. A very simple idea takes care of the problem, reminiscent of FastECPP: we do not need A to be prime, only that $B^2 \equiv N \pmod{A}$ be easy to solve. We consider $A = \prod p_i \approx \sqrt{2N}/M$ for some distinct small primes p_i such that $\left(\frac{N}{p_i}\right) = 1$, with precomputed squares root $B_i^2 \equiv N \pmod{p_i}$. Then we recover B by Chinese remaindering.

A final very important practical improvement are the Large Prime variations (Single, Double, etc.), reminiscent of the B_2 phase in $p-1$ and ECM. We now maintain a database of relations which are smooth, but for a single Large Prime (or up to *a few* large primes). If we hit another relation which is almost smooth but for the *same* large primes, we can combine them and get new relations; or at least get relations involving fewer large primes. Due to the birthday paradox, we expect to find quite a few new relations this way. This idea can be used in all the sieving factorization algorithms.

In practice, the Single and Double large prime variations are easy to implement, but the combinatorics and the costs of handling the associated graphs become quickly horrendous as we allow more of them.

4.3.5 The Special Number Field Sieve

The term "Number Field Sieve" can refer to one of two algorithms. The Special Number Field Sieve only works for numbers of the form $N = r^e - s$, with r, $|s|$ small. The General Number Field Sieve was a later extension of this algorithm to arbitrary integers. We will describe roughly both algorithms, starting with the Special Field Sieve in this subsection, followed by the General Number Field Sieve in the next subsection.

Let us give the general idea of the algorithm. In the Number Field Sieve, we pick a monic irreducible polynomial $f \in \mathbb{Z}[X]$, and an element $m \in \mathbb{Z}/N\mathbb{Z}$ such that $f(m) \equiv 0 \bmod N$. The choice of f and m is as follows. We take

$$d \approx \left(\frac{3\log N}{2\log\log N}\right)^{1/3}.$$

Now select $k \in \mathbb{Z}_{>0}$ which is minimal with respect to $kd \geq e$. Therefore $r^{kd} \equiv sr^{kd-e} \bmod N$. Set

$$m = r^k \quad \text{and} \quad c = sr^{kd-e}.$$

Then $m^d \equiv c \bmod N$. Set

$$f(x) = x^d - c.$$

We can assume f irreducible (which is highly probable), otherwise this either leads directly to a nontrivial factorization of N, or we can replace f by an irreducible factor such that $f(m) \equiv 0 \bmod N$ still holds. Let α be a complex root of f, and note that there is a homomorphism of rings

$$\phi : \mathbb{Z}[\alpha] \simeq \mathbb{Z}[X]/(f) \to \mathbb{Z}/N\mathbb{Z} \quad \text{with} \quad \phi(\alpha) = \overline{m}.$$

For ease of exposition, we assume that $\mathbb{Z}[\alpha]$ is a PID, but it is possible to drop this assumption. We put

$$K = \mathbb{Q}(\alpha),$$

and denote by $N_{K/\mathbb{Q}}$ the norm of K. We choose smoothness bounds B_1 and B_2. Empirically a good choice for B_1 and B_2 is

$$B_1 \approx B_2 \approx \exp((2/3)^{2/3}(\log N)^{1/3}(\log\log N)^{2/3}).$$

Define a set $S = S_1 \cup S_2 \cup S_3$ where

$$S_1 = \{p \in \mathbb{Z};\ p \text{ prime },\ p \leq B_1\},$$

$$S_2 = \text{a set of generators of the unit group} \mathbb{Z}_K^*$$

and

$$S_3 = \{\beta = a + b\alpha \in \mathbb{Z}[\alpha];\ |N_{K/\mathbb{Q}}(\beta)| = p < B_2,\ p \text{ prime }\},$$

where the β are not pairwise associated. Now we set the factor base in $\mathbb{Z}/N\mathbb{Z}$ as

$$\mathcal{F} = \{a_j = \phi(j) \in \mathbb{Z}/N\mathbb{Z};\ j \in S\}.$$

We look for pairs of relatively prime integers (a, b) with $b > 0$, such that $a + bm \in \mathbb{Z}$ is B_1-smooth and $a + b\alpha \in \mathbb{Z}[\alpha]$ has a norm which is B_2-smooth. Note that $N_{K/\mathbb{Q}}(a+b\alpha) = a^d - c(-b)^d$. Such an $a+b\alpha$ is uniquely expressible as a product of elements of $S_2 \cup S_3$. Then, applying ϕ to the factorization of $a + b\alpha$, and setting that equal to the factorization of $a + b\overline{m}$ obtained thanks to S_1, we get a multiplicative relation among the elements in the factor base $\mathcal{F}$ in $\mathbb{Z}/N\mathbb{Z}$. More precisely, if

$$a + b\alpha = \prod_{u \in S_2} u^{\lambda_u} \prod_{g \in S_3} g^{\mu_g}$$

and

$$\phi(a + b\alpha) = \prod_{p \in S_1} p^{v_p}$$

we have

$$\prod_{u\in S_2}\phi(u)^{\lambda_u}\prod_{g\in S_3}\phi(g)^{\mu_g}\equiv\prod_{p\in S_1}p^{v_p}\bmod N.$$

With enough such pairs (a,b), (in fact more than $\#S_1+\#S_2+\#S_3$), we can proceed as in the quadratic sieve to find a factorization of N by doing Gaussian elimination over $\mathbb{F}_2$.

Remark 4.56. The condition $\gcd(a,b)=1$ explains the choice of S_3, because (under the hypothesis $\mathbb{Z}[\alpha]$ PID) if a prime ideal $\mathfrak{p}$ contains an element $a+b\alpha$ with $\gcd(a,b)=1$ then $\mathfrak{p}$ is a first-order prime ideal (its norm is prime). Moreover, it is easy to see that the set of first-order prime ideals of norm p is in one-to-one correspondence with the set of solutions $c \bmod p$ to $f(c)\equiv 0 \bmod p$, with the map given by letting c be the image of α in the mod-$\mathfrak{p}$ reduction map. We can write (p,c) for such ideals and

$$a+b\alpha\in(p,c)\Longleftrightarrow a+bc\equiv 0\bmod p.$$

Hence, it is easy to give a prime factorization of $(a+b\alpha)$ in $\mathbb{Z}[\alpha]$.

Remark 4.57. Like above we can take other bounds B_1' and B_2' for smoothness except for at most one additional prime. For simplicity we do not take such additional bounds here.

Example 4.58. Let us give a famous example. In 1903, A.E. Western found the prime factor

$$2424833=37\cdot 2^{16}+1$$

of

$$F_9=2^{2^9}+1$$

and in 1967, Brillhard determined that $F_9/2424833$ (148 decimal digits) is composite. Let us take $d=5$ (see formula above), $k=103$, $m=2^{103}$,

$$f(x)=x^5+8$$

with root $\alpha=-2^{3/5}$, and work in $\mathbb{Z}[\alpha]=\mathbb{Z}[2^{1/5}]$ which is a PID (here $K=\mathbb{Q}(2^{1/5})$). With this choice and

$$S_1=\{p \text{ prime};\ p\le 1295377\},$$

$$S_2=\{-1,-1+2^{1/5},-1+2^{2/5}-2^{3/5}+2^{4/5}\}$$

and

$$S_3=\{\beta\in\mathbb{Z}[\alpha];\ |N_{K/\mathbb{Q}}(\beta)|=p\le 1294973,\ p\text{ prime}\},$$

Lenstra and al. (with 700 computers at work during 4 months) found

$$F_9 = 2424833 \cdot q_{49} \cdot q_{99},$$

with

$$q_{49} = 7455602825647884208337395736200454918783366342657$$

and

$$q_{99} = 741640062627530801524787141901937474059940781097519023905821316144415759504705008092818711693940737.$$

4.3.6 The General Number Field Sieve

The General Number Field Sieve is an extension of the previous ideas to general integers (i.e. integers not necessarily of the form $r^e - s$ with r, $|s|$ small). The algorithm starts as before, using the same degree d of the extension. The choice of m and f is somewhat different. We pick m by $m = \lfloor N^{1/d} \rfloor$, and we write N in base m with $N = a_d m^d + \cdots + a_0$, $0 \leq a_i < m$. Then f is defined by $f(x) = a_d x^d + \cdots + a_0$. Note that, as required, $f(m) = N \equiv 0 \bmod N$. Unlike the fs generated for the Special Number Field Sieve, the above polynomials may have large coefficients (as high as $N^{1/d}$) and large discriminants, and consequently the number fields generated may be very hard to perform computations in. In particular, attempting to find generators for the units and for prime ideals through exhaustive search, would take too long, and even storing these elements explicitly would take up too much space.

A good solution to these problems is to give up keeping track of explicit factorizations, and just focus on generating a pair (a, b) with $a+bm$ a perfect square in $\mathbb{Z}$ and $a + b\alpha$ a perfect square in $\mathbb{Z}[\alpha]$. Once this pair is generated, we compute the square roots of $a+bm$ and $a+b\alpha$. The principal advantage of this approach is that by not having to keep track of explicit factorizations, we do not need to write down a set of generators of the units of $\mathbb{Z}[\alpha]$, neither a generator for each first-order prime ideal of $\mathbb{Z}[\alpha]$. The principal disadvantage is that one must compute the square root of the (generally) large algebraic integer $a + b\alpha$, one of the most difficult parts of the algorithm. We will not explain here, by lack of time, the method of "quadratic characters", perhaps the most elegant and important aspect of the algorithm, which is used to generate squares in $\mathbb{Z}[\alpha]$ without knowing their complete factorizations.

However, we will mention one small detail of the modified algorithm. It may be the case that a product of numbers of the form $a+b\alpha$ will be a perfect square in $\mathbb{Z}_K$, with $K = \mathbb{Q}(\alpha)$, but not in $\mathbb{Z}[\alpha]$. As it turns out, multiplying the product by $f'(\alpha)^2$ gives a perfect square in $\mathbb{Z}[\alpha]$. If we correspondingly multiply the product of $a + bm$ by $f'(m)^2$, the algorithm can run as before, but with no risk of producing a square root in $\mathbb{Z}_K \setminus \mathbb{Z}[\alpha]$. The only thing to check is that $\gcd(f'(m), N) = 1$: by definition of f , $1 < f(m) < N$, so either this holds, or we have found a factor of N.

Chapter 5

Algebraic Number Theory

5.1 Introduction and definitions

A *number field* K is a finite extension of $\mathbb{Q}$; we may write $K = \mathbb{Q}(x)$ for some x in K (primitive element theorem). All the elements of K are algebraic over $\mathbb{Q}$. An *algebraic integer* in K is an element $x \in K$ satisfying one of the following equivalent properties:

1. the minimal polynomial of x belongs to $\mathbb{Z}[X]$,
2. there exists $Q \in \mathbb{Z}[X]$, Q monic, such that $Q(x) = 0$,
3. $\mathbb{Z}[x]$ is a $\mathbb{Z}$-module of finite type,
4. there exists $M \subset K$ a $\mathbb{Z}$-module of finite type containing $\mathbb{Z}[x]$.

The set of algebraic integers $\mathbb{Z}_K \subset K$ is a ring, which is the proper analog of $\mathbb{Z}_{\mathbb{Q}} = \mathbb{Z} \subset \mathbb{Q}$ for the arithmetic of K.

A number field K has $\dim_{\mathbb{Q}} K = r_1 + 2r_2$ field embeddings $\sigma : K \hookrightarrow \mathbb{C}$. Among them, r_1 embeddings have image contained in $\mathbb{R}$, and $2r_2$ further pairwise conjugate embeddings. The *norm*, *trace*, *characteristic polynomial* of x in K is the determinant, trace, characteristic polynomial of the multiplication by x seen as a $\mathbb{Q}$-linear endormorphism of K. In particular, $N : K^* \to \mathbb{Q}^*$ and $\mathrm{Tr} : K \to \mathbb{Q}$ are group morphisms (for the multiplicative and additive structure, respectively). Concretely,

$$N(x) = \prod_{\sigma : K \hookrightarrow \mathbb{C}} \sigma(x), \quad \mathrm{Tr}(x) = \sum_{\sigma : K \hookrightarrow \mathbb{C}} \sigma(x), \quad \mathrm{Char}_x(T) = \prod_{\sigma : K \hookrightarrow \mathbb{C}} \big(T - \sigma(x)\big).$$

The norm, trace and characteristic polynomial of an algebraic integer are integral. In particular, the units $\mathbb{Z}_K^*$ in $\mathbb{Z}_K$ have norm ± 1.

We consider
$$K \otimes \mathbb{R} \simeq_{\mathbb{R}\text{-algebra}} \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$$
as a Euclidean space, endowed with the canonical Euclidean form $T_2(x) := \sum_{\sigma: K \hookrightarrow \mathbb{C}} |\sigma(x)|^2$. The *discriminant* Δ_K of K is the discriminant of the lattice $(\mathbb{Z}_K, T_2)$. Algebraically, it is the absolute value of the determinant of the matrix $(\mathrm{Tr}(w_i w_j))$, where $\mathbb{Z}_K = \langle w_1, \dots, w_n \rangle_{\mathbb{Z}}$.

An *integral ideal* is a non-zero ideal of $\mathbb{Z}_K$, a *fractional ideal* is a sub $\mathbb{Z}_K$-module of K of rank 1 (equivalent definition: $\mathfrak{A}$ is a fractional ideal if and only if $d\mathfrak{A}$ is integral for some $d \in \mathbb{Z}_{>0}$). Since $\mathbb{Z}_K$ is a Dedekind domain, the fractional ideals form a group and every fractional ideal can be written uniquely as a product of maximal ideals:
$$\mathfrak{A} = \prod_{\mathfrak{p}} \mathfrak{p}^{v_{\mathfrak{p}}(\mathfrak{A})},$$
where $v_{\mathfrak{p}}(\mathfrak{A}) = 0$ for all but finitely many $\mathfrak{p}$. (Note that we exclude the 0 ideal.) A maximal ideal $\mathfrak{p}$ contains a unique prime number p (the generator of $\mathfrak{p} \cap \mathbb{Z}$), and the quotient $\mathbb{Z}_K/\mathfrak{p}$ is a finite field of characteristic p.

A fractional ideal is *principal* if it is of the form $(\alpha) := \alpha\mathbb{Z}_K$ for some $\alpha \in K^*$. The norm $N : K \to \mathbb{Q}$ generalizes to the group of principal ideals by $N(x\mathbb{Z}_K) := |N(x)|$. (It wouldn't be well-defined without the absolute value: x is defined up to units, and units may have norm -1.) This extends to a multiplicative function on the whole group of fractional ideals. For an integral ideal $\mathfrak{A}$, we have $N\mathfrak{A} = \#(\mathbb{Z}_K/\mathfrak{A})$.

The class group of $\mathrm{Cl}(\mathbb{Z}_K)$ is the quotient of the group of fractional ideals by the subgroup of principal ideals. It is a *finite* abelian group. In fact, a simple application of Minkowski's theorem proves that each ideal class contains an integral ideal of norm $O(\sqrt{\Delta_K})$.

A *place* of K is an equivalence class of non-trivial absolute values on K. Concretely, canonical representatives for these classes are given as follows:

- r_1 real places : $|x|_\sigma := |\sigma(x)|$ where $\sigma : K \hookrightarrow \mathbb{R}$ is a real embedding.
- r_2 complex places : $|x|_\sigma := |\sigma(x)|$ where $\sigma : K \hookrightarrow \mathbb{C}$ runs through a system of non-congugate complex embeddings.
- one finite place for each maximal ideal $\mathfrak{p}$: $|x|_{\mathfrak{p}} := N\mathfrak{p}^{-v_{\mathfrak{p}}(x)}$.

The $r_1 + r_2$ real and complex places are called *infinite* places. Given a finite set of places S containing all infinite places, we define the S-integers as
$$\mathbb{Z}_{K,S} = \{x \in K \colon v_{\mathfrak{p}}(x) \geq 0 \text{ for all } x \notin S\}.$$

The subgroup of invertible elements is

$$\mathbb{Z}_{K,S}^* = \{x \in K\colon v_{\mathfrak{p}}(x) = 0 \text{ for all } x \notin S\}\,.$$

This is a $\mathbb{Z}$-module of finite type, a direct product of a (finite) cyclic subgroup $\mu(K)$ containing all roots of unity in K and a free module of rank $\#S - 1$. If S contains only the places at infinity, we obtain the ordinary integers and units in K. Note that $\mathbb{Z}_{K,S}$ is still a Dedekind ring (not of finite type over $\mathbb{Z}$), and we can also define a notion of S-class group, by quotienting the $\mathbb{Z}_{K,S}$-fractional ideals by principal ones. It turns out that $\mathrm{Cl}(\mathbb{Z}_{K,S}) \simeq \mathrm{Cl}(\mathbb{Z}_K)/\left\langle \mathfrak{p}\colon \mathfrak{p} \in S\right\rangle$.

Computational algebraic number theory concerns itself with computing and handling all these objects effectively.

5.2 Concrete representations

We assume that K is given abstractly by the minimal polynomial of a generating element: $K = \mathbb{Q}[X]/(T)$, where T is irreducible in $\mathbb{Q}[X]$. It is no loss of generality to assume that T is integral of degree $n = r_1 + 2r_2$. We shall furthermore assume that T is monic. (We can reduce to this case by a change of variable, although most algorithms can be adapted to work directly with non-monic inputs, more efficiently than on the transformed monic polynomial.) We will not explicitly evaluate the complexity of most of our algorithms computing invariants of K, but their input size would be $n \log ||T||_\infty$, and we hope to obtain runtimes bounded by a polynomial in this input size. (We shall not be successful.)

We may then work in K either as in any (univariate) polynomial quotient ring, or as a $\mathbb{Q}$-vector space with canonical basis $1, X, \dots, X^{n-1}$. In particular, this yields a direct way to compute the norm, trace and characteristic polynomial of $x \in K$ using $\mathbb{Q}$-linear algebra.

The $r_1 + 2r_2$ embeddings are given by $\sigma : X \mapsto \alpha_\sigma$, where the α_σ are the complex roots of T. Since we know how to approximate the complex roots of T within a guaranteed fixed accuracy, we may approximate any $\sigma(x) \in \mathbb{C}$ as floating point complex numbers to an arbitray fixed precision. This gives a different, analytic, method to compute norms, traces and characteristic polynomials (bound denominators, approximate, then round).

$\mathbb{Z}_K$ and all fractional ideals are free $\mathbb{Z}$-module of rank n and can be represented by a $\mathbb{Z}$-basis. Once we fix a $\mathbb{Z}$-basis $(w_1, \dots, w_n)$ for $\mathbb{Z}_K$, any integral ideal has a canonical basis, given by the Hermite Normal Form, i.e. of the form $(w_i)H$ where H is a square matrix in HNF. Note that $\det H$ is the index of the submodule, hence the norm of the ideal. The product $\mathfrak{A} \times \mathfrak{B}$

is the $\mathbb{Z}$-module generated by all n^2 products a_ib_i of the generators of $\mathfrak{A}$ and $\mathfrak{B}$. The quotient $\mathfrak{A}\mathfrak{B}^{-1}$ is $(\mathfrak{A} : \mathfrak{B})$, where

$$(A : B) := \{\alpha \in K, \alpha B \subset A\},$$

for any $\mathbb{Z}$-modules A and B. Given $\mathbb{Z}$-bases, this quotient can also be computed using $\mathbb{Z}$-linear algebra (i.e. the HNF algorithm).

Units and ideal classes are represented in the obvious way: as elements of K and by any (integral) ideal representative respectively. It is not yet clear how to compute with ideal classes, nor how to enumerate a complete system of representatives. Analogousy, if $\mu(K)$ is relatively straightforward (amounts to factoring cyclotomic polynomials over K, which can be done using generalizations of the techniques we saw over $\mathbb{Q}$), how to find non-torsion units?

5.3 The maximal order $\mathbb{Z}_K$, first steps

An *order* of K is a subring $\mathcal{O}$ (in particular, containing 1) whose field of fractions is K; this last condition is equivalent to $\text{rank}_{\mathbb{Z}}\mathcal{O} = \dim_{\mathbb{Q}} K$. Note that $\mathbb{Z}_K$ is an order and that all orders are contained in $\mathbb{Z}_K$ (property 4), with finite index. We define the discriminant $\Delta_{\mathcal{O}}$ of $\mathcal{O}$, from any $\mathbb{Z}$-basis of $\mathcal{O}$, by mimicking the definition of $\Delta_K = \Delta_{\mathbb{Z}_K}$. A direct computation involving a van der Monde matrix shows that if x is an algebraic integer generating K, with minimal polynomial T, then $\Delta_{\mathbb{Z}[x]} = \text{disc}T := \text{Res}(T, T')$.

The main point of orders is that

- they are easy to construct: $\mathbb{Z}[x]$ is an order for any $x \in \mathbb{Z}_K$, in particular. In particular, given $T \in \mathbb{Z}[X]$ monic such that $K = \mathbb{Q}[X]/(T)$, we obtain a canonical *equation order* (generated by the class of X).

- they approximate $\mathbb{Z}_K$. In fact they have finite index in $\mathbb{Z}_K$, which is easy to bound since we have $\Delta_{\mathcal{O}} = [\mathbb{Z}_K : \mathcal{O}]^2 \Delta_K$.

- they are somewhat imperfect compared to the maximal order, but still have interesting arithmetic properties: they are noetherian integral domains of dimension 1. But a non-maximal order is not integrally closed and the norm $\mathcal{O}/\mathfrak{A}$ is no longer multiplicative, finitely many maximal ideals are not invertible, etc.

Remark 5.1. More generally, orders appear as "rings of stabilizers". For instance, if $A \subset K$ is a $\mathbb{Z}$-module of rank n, it is easy to see that $\mathcal{O} = (A : A)$ is an order.

In many applications (e.g. factoring polynomials in $K[X]$, splitting primes in K) it is enough to know any order, the computation being more efficient if its index is reasonably small. For instance, given a $\mathbb{Z}$-basis (w_i) for an order $\mathcal{O}$, any integer of K is a $\mathbb{Q}$-linear combination of the w_i, with denominator bounded by the exponent of the additive group $\mathbb{Z}_K/\mathcal{O}$ (itself obviously bounded by the index). On the other hand, to compute more subtle invariants like the class groups and units we will need $\mathbb{Z}_K$ itself.

Definition 5.2. Let m be an integer. An order $\mathcal{O} \subset \mathbb{Z}_K$ is *m-maximal* if m and the index $[\mathbb{Z}_K : \mathcal{O}]$ are coprime.

Theorem 5.3 (Zassenhaus)**.** *Given an order $\mathcal{O}$ and a prime p, we can find in polynomial time a basis for an order $\mathcal{O}_p \supset \mathcal{O}$ such that $\mathcal{O}_p$ is p-maximal.*

To prove this theorem we need some lemmas.

Lemma 5.4. *Let $I_p = \mathrm{Rad}(p\mathcal{O})$ the radical of $p\mathcal{O}$:*

$$I_p = \{x \in \mathcal{O};\ \exists m \geq 1,\ x^m \in p\mathcal{O}\}.$$

We have:

1. *I_p is an ideal of $\mathcal{O}$.*

2. *We have*
$$I_p = \prod_{1 \leq i \leq g} \mathfrak{p}_i,$$
the product being over all distinct prime ideals $\mathfrak{p}_i$ of $\mathcal{O}$ which lie above p.

3. *There exists an integer m such that $I_p^m \subset p\mathcal{O}$.*

Proof. For (1) it is sufficient to see that if $x^m \in p\mathcal{O}$ and $y^n \in p\mathcal{O}$, then $(x+y)^{m+n} \in p\mathcal{O}$ (binomial theorem).

For (2), if $\mathfrak{p}_i$ lies above p, we have $p\mathcal{O} \subset \mathfrak{p}_i$, so that

$$\begin{aligned} x \in I_p &\implies x^m \in p\mathcal{O} \subset \mathfrak{p}_i \text{ for some } m \\ &\implies x \in \mathfrak{p}_i \text{ (because } \mathfrak{p}_i \text{ is prime)} \\ &\implies x \in \bigcap \mathfrak{p}_i = \prod \mathfrak{p}_i, \end{aligned}$$

since the distinct maximal ideals $\mathfrak{p}_i$ are pairwise coprime.

Conversely suppose that $x \in \prod \mathfrak{p}_i$. The set of ideals of $\mathcal{O}$ containing $p\mathcal{O}$ is in one-to-one correspondence with the ideals of $R = \mathcal{O}/p\mathcal{O}$. In particular they are in finite number and if α is the class of x in R, the $\alpha^n R$ are in finite number and there exist integers n and $k \geq 1$ such that

$$\alpha^n R = \alpha^{n+k} R.$$

This implies that

$$\alpha^n(1 - \alpha^k \beta) = 0$$

for some $\beta \in R$. Since for every i, $\alpha \in \overline{\mathfrak{p}_i}$, we have $\alpha^k \in \overline{\mathfrak{p}_i}$ and necessarily

$$1 - \alpha^k \beta \notin \overline{\mathfrak{p}_i} \quad \text{for every } i$$

(if not $1 \in \overline{\mathfrak{p}_i}$ which is a maximal ideal of R). We deduce from this that $(1 - \alpha^k \beta)R = R$ and that $1 - \alpha^k \beta$ is invertible in R. Finally, since $\alpha^n(1 - \alpha^k \beta) = 0$ we have $\alpha^n = 0$ so that $x^n \in p\mathcal{O}$ which implies $x \in I_p$ by definition of I_p. □

Lemma 5.5. *Let $\mathcal{O}' := (I_p : I_p)$. The order $\mathcal{O}$ is p-maximal if and only if $\mathcal{O} = \mathcal{O}'$. Otherwise, $p \mid [\mathcal{O}' : \mathcal{O}] \mid p^n$.*

Proof. Since I_p is an ideal, it is easy to see that $\mathcal{O}'$ is an order containing $\mathcal{O}$ (see also Remark 5.1). As $p \in I_p$ we have by definition of $\mathcal{O}'$

$$x \in \mathcal{O}' \Longrightarrow xp \in I_p \subset \mathcal{O},$$

from what we deduce that

$$p\mathcal{O}' \subset \mathcal{O} \subset \mathcal{O}'.$$

It follows that

$$[\mathcal{O}' : \mathcal{O}] \mid p^n$$

and that $\mathcal{O}' = \mathcal{O}$ if O is p-maximal ($[\mathbb{Z}_K : \mathcal{O}] = [\mathbb{Z}_K : \mathcal{O}'][\mathcal{O}' : \mathcal{O}]$ so that if p does not divide $[\mathbb{Z}_K : \mathcal{O}]$, then $[\mathcal{O}' : \mathcal{O}] = 1$).

Conversely suppose that $\mathcal{O}' = \mathcal{O}$. Let

$$R = \{x \in \mathbb{Z}_K;\ \exists j \geq 1,\ p^j x \in \mathcal{O}\}.$$

R is an order containing $\mathcal{O}$. Moreover it is p-maximal: in fact, if $p \mid [\mathbb{Z}_K : R]$, there exists an $x \in \mathbb{Z}_K$ such that $x \notin R$ and $px \in R$ which contradicts the definition of R. As an order, R is finitely generated over $\mathbb{Z}$ and there exists $r \geq 1$ such that

$$p^r R \subset \mathcal{O}$$

(take the max of j such that $p^j x_i \in \mathcal{O}$ for a finite generating set (x_i) of R).

Since $I_p^m \subset p\mathcal{O}$ for some m (previous lemma) we have

$$RI_p^{mr} \subset \mathcal{O}.$$

By contadiction, suppose that $\mathcal{O}$ is not p-maximal, so that $R \neq \mathcal{O}$. Then we have $R \not\subset \mathcal{O}$. Let n be the largest index such that $RI_p^n \not\subset \mathcal{O}$. By assumption $1 \leq n < mr$ and

$$RI_p^{n+1} \subset \mathcal{O}.$$

Choose an $x \in RI_p^n \setminus \mathcal{O}$. The last inclusion implies that $xI_p \subset \mathcal{O}$. Since

$$RI_p^{n+m+1} \subset I_p^m \subset p\mathcal{O}$$

it follows that if $y \in I_p$ we have $(xy)^{n+m+1} \in p\mathcal{O}$ and finally $xy \in I_p$. This implies $xI_p \subset I_p$ and we have $x \in \mathcal{O}'$. Contradiction because $x \notin \mathcal{O}$ and by assumption $\mathcal{O}' = \mathcal{O}$. □

Proof. (of Theorem 5.3)
The algorithm is now obvious: compute $I_p/p\mathcal{O}$, lift to I_p, then compute $\mathcal{O}'$. Either $\mathcal{O} = \mathcal{O}'$ is p-maximal or we replace $\mathcal{O}$ by $\mathcal{O}'$ and restart, dividing the index at least by p. □

Corollary 5.6. *Given the primes p such that $p^2 \mid \mathrm{disc} T$, we can compute a $\mathbb{Z}$-basis for $\mathbb{Z}_K$ in polynomial time.*

Proof. Recall that we have

$$\mathrm{disc}(T) = \mathrm{disc}(K)[\mathbb{Z}_K : \mathbb{Z}[\theta]]^2,$$

so that if $p^2 \nmid \mathrm{disc} T$, $\mathbb{Z}[\theta]$ is p-maximal. Otherwise, compute $\mathcal{O}_p$ for all those given p and return $\sum \mathcal{O}_p$ (HNF). □

Interestingly, Zassenhaus's algorithm still works if p is only assumed to be squarefree. Either the algorithm exhibit a zero divisor in $\mathbb{Z}/p\mathbb{Z}$ (from which we can factor p and restart), or it produces a p-maximal order. Using these ideas one can prove a stronger result:

Theorem 5.7 (Buchmann-Lenstra)**.** *There are polynomial time algorithm that given a number field K and one of 1), 2) below determines the other:*

1. *the ring of integers of K,*
2. *the largest squarefree divisor of Δ_K.*

Finding the largest squarefree divisor of a given integer is currently essentially as hard as full integer factorization, hence computing $\mathbb{Z}_K$ is difficult; but it becomes easy if an explicit factorization is given. To see why 2) is at least as hard as 1), consider the simplest case of a quadratic field $K = \mathbb{Q}(\sqrt{D})$ for some integer D. How would you compute $\mathbb{Z}_K$ without assuming that D is squarefree?

5.4 Dedekind's criterion and the general algorithm

A very important byproduct of Zassenhaus's algorithm is that it is trivial to check whether a given order $\mathcal{O}$ is p-maximal for p prime (or squarefree, using Buchmann and Lenstra's trick). The recipe simplifies if $\mathcal{O}$ is the equation order:

Theorem 5.8 (Dedekind)**.** *Let p be a prime number. Let $K = \mathbb{Q}(\theta)$, and $T \in \mathbb{Z}[X]$ be the monic, minimal polynomial of θ. Suppose that*

$$T \equiv \prod_i P_i^{e_i} \pmod{p\mathbb{Z}[X]},$$

where the P_i are monic, irreducible and distinct modulo p. Let

$$g := \prod P_i, \quad h := \prod P_i^{e_i - 1}, \quad f := (T - gh)/p \in \mathbb{Z}[X].$$

1. *Then $\mathbb{Z}[\theta]$ is p-maximal if and only if $\gcd(\overline{f}, \overline{g}, \overline{h}) = 1$ in $\mathbb{F}_p[X]$.*

2. *Moreover let $\mathcal{O}' = (I_p : I_p)$ be defined as above when we start with $\mathcal{O} = \mathbb{Z}[\theta]$. If U is a monic lift of $\overline{T}/\gcd(\overline{f}, \overline{g}, \overline{h})$ to $\mathbb{Z}[X]$, we have*

$$\mathcal{O}' = \mathbb{Z}[\theta] + \frac{1}{p} U(\theta)\mathbb{Z}[\theta]$$

and if $m = \deg \gcd(\overline{f}, \overline{g}, \overline{h})$ then $[\mathcal{O}' : \mathbb{Z}[\theta]] = p^m$ hence $\mathrm{disc}\mathcal{O}' = \mathrm{disc}T/p^{2m}$.

Proof. See exercises. □

Corollary 5.9. *An Eisenstein polynomial at p yields a p-maximal equation order.*

Proof. $f = X$, $g = X^{n-1}$, $h = (T - X^n)/p$. The gcd is 1. □

Now the algorithm can be modified. We can use directly Dedekind's criterion in the algorithm given in Corollary 5.6. But we can also do as follows. We compute $\mathbb{Z}_K$ by successive enlargements from $\mathcal{O} = \mathbb{Z}[\theta]$. For every prime p such that $p^2 \mid \mathrm{disc}(T)$ we proceed as follows:

1. By using Dedekind's criterion, we check whether $\mathcal{O}$ is p-maximal.

2. If it is not, we enlarge it thanks to the second part of Dedekind's criterion.

3. If the new discriminant is not divisible by p^2, then we are done, otherwise we compute $\mathcal{O}'$ as described in the Zassenhaus theorem.

4. If $\mathcal{O}' = \mathcal{O}$ then $\mathcal{O}$ is p-maximal and we move on to the next prime, if any (here again we can start by Dedekind's criterion because if $\mathbb{Z}[\theta]$ is not p-maximal, our $\mathcal{O}$ is also).

5. Otherwise we replace $\mathcal{O}$ by $\mathcal{O}'$ an use again Zassenhaus.

This algorithm is known as the *round 2* algorithm.

5.5 Splitting of primes

Theorem 5.10 (Kummer). *Let $K = \mathbb{Q}(\theta)$, $T \in \mathbb{Z}[X]$ the monic minimal polynomial of θ. If $\mathbb{Z}[\theta]$ is p-maximal, the factorization of T mod p mirrors the factorization of $p\mathbb{Z}_K$. More precisely, if*

$$T \equiv \prod_{i=1}^{g} P_i^{e_i} \pmod{p\mathbb{Z}[X]},$$

where the P_i are monic, irreducible and distinct modulo p. Then

$$p\mathbb{Z}_K = \prod_i \mathfrak{p}_i^{e_i},$$

where the $\mathfrak{p}_i := p\mathbb{Z}_K + P_i(\theta)\mathbb{Z}_K$ are distinct maximal ideals, with residual degree $\deg P_i$.

Proof. Set $f_i = \deg P_i$. Let us assume that we have proved the following lemma.

Lemma 5.11. *We have:*

1. *For all i, either $\mathfrak{p}_i = \mathbb{Z}_K$, or $\mathbb{Z}_K/\mathfrak{p}_i$ is a field of cardinality p^{f_i}.*

2. *If $i \neq j$ then $\mathfrak{p}_i + \mathfrak{p}_j = \mathbb{Z}_K$.*

3. *$p\mathbb{Z}_K \mid \prod \mathfrak{p}_i^{e_i}$.*

After reordering the $\mathfrak{p}_i$ we can assume that $\mathfrak{p}_i \neq \mathbb{Z}_K$ for $i \leq s$ and $\mathfrak{p}_i = \mathbb{Z}_K$ for $s < i \leq g$. Then by Lemma 5.11 (1), the $\mathfrak{p}_i$ are prime for $i \leq s$ and since they contain $p\mathbb{Z}_K$ they are above p. Lemma 5.11 (1) also implies that the f_i

$(i \leq s)$ are the residual indices of $\mathfrak{p}_i$. By (2) we know that the $\mathfrak{p}_i$ for $i \leq s$ are distinct and (3) implies that

$$p\mathbb{Z}_K = \prod_{i=1}^{s} \mathfrak{p}_i^{d_i}$$

where $d_i \leq e_i$. Hence we have $n = \sum d_i f_i$. Since we have also $n = \deg T = \sum e_i f_i$ it follows that $s = g$ and $e_i = f_i$ for all i. □

Proof. (of Lemma 5.11)
(1): Set $K_i = \mathbb{F}_p[X]/(P_i)$. It is a field of cardinality p^{f_i}.Thus it is sufficient to prove that either $\mathfrak{p}_i = \mathbb{Z}_K$ or $\mathbb{Z}_K/\mathfrak{p}_i \simeq K_i$. Now it is easy to see that $\mathbb{Z}[X]/(p, P_i) \simeq K_i$, hence (p, P_i) is a maximal ideal of $\mathbb{Z}[X]$. But the kernel of the homomorphism ϕ from $\mathbb{Z}[X]$ to $\mathbb{Z}_K/\mathfrak{p}_i$ which sends X to $\theta \bmod \mathfrak{p}_i$ contains this ideal, hence is either $\mathbb{Z}[X]$ or (p, P_i). If we show that ϕ is onto, this will imply $\mathfrak{p}_i = \mathbb{Z}_K$ or $\mathbb{Z}_K/\mathfrak{p}_i \simeq \mathbb{Z}[X]/(p, P_i) \simeq K_i$, proving (1). But, saying ϕ is onto means that $\mathbb{Z}_K = \mathbb{Z}[\theta] + \mathfrak{p}_i$. Since $p\mathbb{Z}_K \subset \mathfrak{p}_i$ we have

$$[\mathbb{Z}_K : \mathbb{Z}[\theta] + \mathfrak{p}_i] \mid [\mathbb{Z}_K : \mathbb{Z}[\theta] + p\mathbb{Z}_K] = \gcd([\mathbb{Z}_K : \mathbb{Z}[\theta]], [\mathbb{Z}_K : p\mathbb{Z}_K]).$$

Since p does not divide the index and since $[\mathbb{Z}_K : p\mathbb{Z}_K] = p^n$, this shows that $[\mathbb{Z}_K : \mathbb{Z}[\theta] + \mathfrak{p}_i] = 1$ hence the surjectivity of ϕ.

(2): since P_i and P_j are coprime in $\mathbb{F}_p[X]$ there exist polynomials such that $UP_i + VP_j - 1 \in p\mathbb{Z}[X]$ so that

$$U(\theta)P_i(\theta) + V(\theta)P_j(\theta) = 1 + pW(\theta)$$

for some $W \in \mathbb{Z}[X]$. This implies $1 \in \mathfrak{p}_i + \mathfrak{p}_j$ and the conclusion.

(3): Set $u_i = P_i(\theta)$. We have

$$\prod \mathfrak{p}_i^{e_i} \subset (p, \prod u_i^{e_i})$$

by distributivity. But this last ideal is in fact $p\mathbb{Z}_K$. Indeed $\supset$ is trivial. Conversely $\prod P_i^{e_i} - T \in p\mathbb{Z}[X]$ hence taking $X = \theta$ we obtain $\prod u_i^{e_i} \in p\mathbb{Z}[\theta] \subset p\mathbb{Z}_K$. □

If Dedekind's criterion tells us that the equation order is *not* p-maximal, we can still compute a p-maximal order $\mathcal{O}$ using Zassenhaus's method (Theorem 5.3). In fact, for simplicity, assume we know $\mathbb{Z}_K$. Then we can compute $I_p = \prod \mathfrak{p}_i$ as in Zassenhaus's method, and finding the $\mathfrak{p}_i$ is equivalent to splitting the separable algebra $\mathbb{Z}_k/I_p$, which can be done using an adaptation of Berlekamp's algorithm. Now what can we say in the general case i.e. when $\mathbb{Z}[\theta]$ is not necessarily p-maximal? We have the following result.

Theorem 5.12. *Let $K = \mathbb{Q}(\theta)$, $T \in \mathbb{Z}[X]$ the monic minimal polynomial of θ. If*

$$T \equiv \prod_i P_i^{e_i} \pmod{p\mathbb{Z}[X]},$$

where the P_i are monic, irreducible and distinct modulo p. Then

$$p\mathbb{Z}_K = \prod_i \mathfrak{a}_i,$$

where the $\mathfrak{a}_i := p\mathbb{Z}_K + P_i^{e_i}(\theta)\mathbb{Z}_K$ are pairwise coprime ideals. Furthermore, if f_i is the degree of P_i we have $N_{K/\mathbb{Q}}(\mathfrak{a}_i) = p^{e_i f_i}$ and all prime ideals dividing $\mathfrak{a}_i$ are of residual degree divisible by f_i.

Proof. See exercises. □

5.6 Determination of $h(K)$, Cl(K) and $\mathbb{Z}_K^*$

Our new goal is to compute the class group of K denoted by $\mathrm{Cl}(K)$ which has cardinality $h(K)$ (the class number of K) and $\mathbb{Z}_K^*$ the group of units of K. Recall that if $\omega(K)$ is the set of roots of unity in K and if (r_1, r_2) is the signature of K we have by Dirichlet:

$$\mathbb{Z}_K^* \simeq \omega(K) \times \mathbb{Z}^{r_1+r_2-1}.$$

Define the *logarithmic embedding* of K^* in $\mathbb{R}^{r_1+r_2}$ by

$$L(x) = (\log|\sigma_1(x)|, \dots, \ln|\sigma_{r_1}(x)|, 2\ln|\sigma_{r_1+1}(x)|, \dots, 2\log|\sigma_{r_1+r_2}(x)|).$$

Then $L(\mathbb{Z}_K^*)$ is a lattice of rank $r := r_1 + r_2 - 1$ in the hyperplane

$$H = \Big\{x \in \mathbb{R}^{r_1+r_2};\ \sum_{1\le i\le r_1+r_2} x_i = 0\Big\}$$

and the kernel of L is $\omega(K)$. The volume of this lattice (the absolute value of the determinant of any $\mathbb{Z}$-basis of the lattice) is called the *regulator* of K and denoted by $R(K)$.

The constants $h(K)$ and $R(K)$ are linked in the following way. Let ζ_K and ζ be the zeta functions of K and $\mathbb{Q}$ respectively and put $w(K) = \#\omega(K)$. Recall that if $\mathrm{Re}(s) > 1$ we have

$$\zeta_K(s) = \sum_{\mathfrak{a}\text{ non zero ideal}} \frac{1}{N(\mathfrak{a})^s} \quad\text{and}\quad \zeta(s) = \sum_{n\ge 1}\frac{1}{n^s}.$$

Then

$$\frac{h(K)R(K)2^{r_1}(2\pi)^{r_2}}{w(K)\sqrt{|\Delta_K|}} = \lim_{s\to 1^+} \frac{\zeta_K(s)}{\zeta(s)},$$

from which we can write:

$$h(K)R(K) = \frac{w(K)\sqrt{|\Delta_K|}}{2^{r_1}(2\pi)^{r_2}} \prod_p \frac{1-\frac{1}{p}}{\prod_{\mathfrak{p}|p}(1-\frac{1}{N(\mathfrak{p})})}.$$

In what follows we shall assume GRH and describe a conjecturally[1] subexponential algorithm due to Cohen, Diaz y Diaz and Olivier (1997)[2]. Note that if we do not assume GRH, the method is still valid, but with very bad bounds (essentially Minkowski's bounds) so that the algorithm becomes exponential.

The algorithm is decomposed in five main steps.

1. Find a system of generators $\overline{g_1}, \ldots, \overline{g_k}$ of $\mathrm{Cl}(K)$ and let $g_1, \ldots, g_k$ be ideals above these generators.

2. Find l (many) relations in $\mathrm{Cl}(K)$ between the $\overline{g_i}$. Write these relations as

   $$\prod_{i=1}^{k} g_i^{m_{i,j}} = \alpha_j \mathbb{Z}_k \ \ (1 \le j \le l)$$

   where $m_{i,j} \in \mathbb{Z}$ and $\alpha_j \in \mathbb{Z}_K$.

3. Let

   $$M = (m_{i,j})_{\substack{1\le i\le k\\ 1\le j\le l}} \quad \text{and} \quad V = (\alpha_j)_{1\le j\le l}.$$

   Perform Hermite and Smith reductions on M and V to obtain a tentative class group and units group. Let $h'(K)$ and $R'(K)$ be the corresponding tentative class number and regulator.

4. By using the analytic class number and regulator formula (see above), check that the product $h'(K)R'(K)$ is correct up to factor 2. If not, find a few more relations and go to step 3.

5. Now $h'(K) = h(K)$ and $R'(K) = R(K)$. Compute a system of fundamental units, output it and the class group.

[1] it has only been proved subexponential in the quadratic case.

[2] The basic ideas of this algorithm are due to Buchmann.

Now let us describe these five steps in detail.

Step 1. Under GRH Bach (1990) has proved that there exists $C > 0$ such that if $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ are the non-inert prime ideals of norm less than $C \log^2 |\Delta_K|$ then $\overline{\mathfrak{p}_1}, \ldots, \overline{\mathfrak{p}_k}$ generate the class group. One can take $C = 6$ for quadratic fields and $C = 12$ otherwise. In practice it is a good thing to take a smaller C and to check that the classes of all other prime ideals of larger norm are in the group generated by the classes of the ones that we have taken. At this point, we have

$$\mathrm{Cl}(K) \simeq \mathbb{Z}^k/\Lambda,$$

where Λ is a lattice of relations. In fact we have the exact sequence

$$0 \longrightarrow \Lambda \longrightarrow \mathbb{Z}^k \longrightarrow \mathrm{Cl}(K) \longrightarrow 0,$$

where the map f from $\mathbb{Z}_K$ to $\mathrm{Cl}(K)$ is given by $f(v_1, \ldots, v_k) = \prod \overline{g_i}^{v_i}$. The next steps consist in finding Λ.

Step 2. We shall use three kinds of relations:

1. Relations of the shape
$$p\mathbb{Z}_k = \prod_{i=1}^{k} \mathfrak{p}_i^{e_{i,p}},$$
for p prime up to a certain bound. These splittings are easy to establish when $p \nmid [\mathbb{Z}_K : \mathbb{Z}[\theta]]$. Otherwise there exists a good method due to Buchmann and Lenstra (1991) to find them.

2. Find $\alpha \in \mathbb{Z}_K$ with small norms and factor α on the factor base of the $\mathfrak{p}_i$:
$$\alpha\mathbb{Z}_K = \prod_{i=1}^{k} \mathfrak{p}_i^{v_{\mathfrak{p}_i}(\alpha)}.$$

3. Generate small random exponents u_i and consider the ideal $I = \prod_{i=1}^{k} \mathfrak{p}_i^{u_i}$. Then LLL-reduce in a random direction[3] this ideal, obtaining an ideal

[3] Let $v = (v_i)_{1 \le i \le n} \in \{x \in \mathbb{R}^n;\ x_{r_2+i} = x_i,\ \forall\, r_1 \le i \le r_1 + r_2\}$. Define

$$\| \alpha \|_v^2 = \sum_{i=1}^{n} e^{v_i} |\sigma_i(\alpha)|^2.$$

A $\mathbb{Z}$-basis $\alpha_1, \ldots, \alpha_n$ is said LLL-reduced in the direction of v if it is LLL-reduced for $\| \cdot \|_v^2$. The interest is that, if α is a minimum for $\| \cdot \|_v^2$ then it is an algebraic minimum of I (if $\beta \in I$ is such that $|\sigma_i(\beta)| < |\sigma_i(\alpha)|$ for every i then $\beta = 0$). Here, even if we have not a minimum, it is sufficient for our purpose.

$J = I/\alpha$. If J factors on the $\mathfrak{p}_i$ as $J = \prod \mathfrak{p}_i^{v_i}$ we obtain

$$\prod_{i=1}^{k} \mathfrak{p}_i^{u_i - v_i} = \alpha \mathbb{Z}_K.$$

Step 3. Put

$$M = (m_{i,j})_{\substack{1 \le i \le k \\ 1 \le j \le l}} \in M_{k \times l}(\mathbb{Z}) \quad \text{and} \quad V = (\alpha_j)_{1 \le j \le l} \in M_{1 \times l}(\mathbb{Z}_K),$$

with $l > k$. We have

$$\alpha_j \mathbb{Z}_K = \prod_{i=1}^{k} g_i^{m_{i,j}}.$$

Consider the complex logarithmic embedding L_C defined by

$$L_C(\alpha) = \left(n_i \Big(\log \sigma_i(\alpha) - \frac{\log N_{K/\mathbb{Q}}(\alpha)}{n} \Big) \right)_{1 \le i \le r_1 + r_2},$$

where log is any determination of the logarithm and where $n_i = 1$ if $i \le r_1$, $n_i = 2$ otherwise. When applied to V it gives a matrix

$$M_C \in M_{(r_1+r_2) \times l}(\mathbb{C}).$$

We perform HNF on M, doing the same operations on M_C. We obtain a matrix

$$\begin{pmatrix}
0 & \cdots & 0 & * & \cdots & \cdots & \cdots & * & * & \cdots & \cdots & \cdots & * \\
\vdots & & \vdots & \vdots & & Z & & \vdots & \vdots & & & & \vdots \\
\vdots & & \vdots & * & \cdots & \cdots & \cdots & * & \vdots & & & & \vdots \\
\vdots & & \vdots & * & \cdots & \cdots & \cdots & * & \vdots & & A & & \vdots \\
\vdots & & \vdots & 0 & \ddots & & B & \vdots & \vdots & & & & \vdots \\
\vdots & & \vdots & \vdots & \ddots & \ddots & & \vdots & \vdots & & & & \vdots \\
\vdots & & \vdots & 0 & \cdots & \cdots & 0 & * & * & \cdots & \cdots & \cdots & * \\
\vdots & & \vdots & 0 & \cdots & & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 \\
\vdots & & \vdots & 0 & \cdots & & \cdots & 0 & 0 & 1 & \ddots & & 0 \\
\vdots & & \vdots & 0 & \cdots & & \cdots & 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \vdots & 0 & \cdots & & \cdots & 0 & \vdots & & \ddots & 1 & 0 \\
0 & & 0 & 0 & \cdots & & \cdots & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}$$

and an $(r_1+r_2)\times l$ complex matrix. If Z has a rows, we have $\mathrm{rank} M = k-a$ but we want $\mathrm{rank} M = k$. So if $a > 0$ we take more relations until $a = 0$. Finally we have put our M in a matrix of the shape

$$\begin{pmatrix}
0 & 0 & * & \cdots & \cdots & \cdots & * & * & \cdots & \cdots & \cdots & * \\
\vdots & \vdots & 0 & \ddots & & B & \vdots & \vdots & & A & & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & & \vdots & \vdots & & & & \vdots \\
\vdots & \vdots & 0 & \cdots & \cdots & 0 & * & * & \cdots & \cdots & \cdots & * \\
\vdots & \vdots & 0 & \cdots & & \cdots & 0 & 1 & 0 & \cdots & \cdots & 0 \\
\vdots & \vdots & 0 & \cdots & & \cdots & 0 & 0 & 1 & \ddots & & 0 \\
\vdots & \vdots & 0 & \cdots & & \cdots & 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & 0 & \cdots & & \cdots & 0 & \vdots & & \ddots & 1 & 0 \\
0 & 0 & 0 & \cdots & & \cdots & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}$$

and have transformed M_C into a matrix

$$\begin{pmatrix}
* & \cdots & * & * & \cdots & \cdots & \cdots & * & * & \cdots & \cdots & \cdots & * \\
\vdots & U_C & \vdots & \vdots & & B_C & & \vdots & \vdots & & A_C & & \vdots \\
* & \cdots & * & * & \cdots & \cdots & \cdots & * & * & \cdots & \cdots & \cdots & *
\end{pmatrix}.$$

The determinant $\det B = h'(K) \neq 0$ is a multiple of the class number and is equal to the class number if and only if $M\mathbb{Z}^k = \Lambda$ the lattice of relations. Moreover each column of U_C is the complex logarithmic embedding of $\alpha \in \mathbb{Z}_K$ such that $\alpha\mathbb{Z}_K = \prod \mathfrak{p}_i^0 = \mathbb{Z}_K$. This shows that $\alpha \in \mathbb{Z}_K^*$. Consider now $U_R = \mathrm{Re}(U_C)$. The $\mathbb{Z}$-lattice F generated by its columns is a sublattice of $L(\mathbb{Z}_K^*)$. We compute $\mathrm{Vol}(F) = R'(K)$ which is a multiple of $R(K)$.

Step 4. In the formula seen above that gives $h(K)R(K)$ if T is the quantity obtained by truncating the Euler product at $N(\mathfrak{p}) < C\log^2|\Delta_K|$ then, still under GRH, we have by a result of Bach-Schoof

$$\frac{h(K)R(K)}{\sqrt{2}} < T < \sqrt{2}h(K)R(K).$$

Now we use the following proposition.

Proposition 5.13. *We have $h'(K) = h(K)$ and $R'(K) = R(K)$ if and only if $h'(K)R'(K) < T\sqrt{2}$.*

Proof. The ⇒ part is obvious. Conversely, we have $h'(K)R'(K) < 2h(K)R(K)$ but $h'(K)$ and $R'(K)$ are multiples of $h(K)$ and $R(K)$. □

So, as long as $h'(K)R'(K) \geq T\sqrt{2}$ we compute more relations as before until we can apply our proposition and finally we obtain the real values of $h(K)$ and $R(K)$. Moreover primes ideals corresponding to the rows of B generate $\mathrm{Cl}(K)$, B giving the relations between them. Columns of U_C corresponding to a basis of U_R are the images by L_C of a system of fundamental units, that we can recover by Gaussian elimination. Applying then SNF on B allows us to obtain the structure of $\mathrm{Cl}(K)$

$$\mathrm{Cl}(K) \simeq \oplus\, \mathbb{Z}/d_i\mathbb{Z}$$

(where $d_{i+1} \mid d_i$) and generators of the cyclic components as products of prime ideals.

Bibliography

[1] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena, *Primes is in P*, Ann. of Math. (2) **160** (2004), no. 2, 781-793.

[2] R. Alford, Andrew Granville, and Carl Pomerance, *There are infinitely many Carmichael numbers*, Ann. of Math. (2) **139** (1994), no. 3, 703-722.

[3] E. Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. **55** (1990), no. 191, 355-380.

[4] E. R. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp. **24** (1970), 713-735.

[5] Peter Borwein and Tamás Erdélyi, *Polynomials and polynomial inequalities*, Graduate Texts in Mathematics, vol. 161, Springer-Verlag, New York, 1995.

[6] H. Cohen, *A course in computational algebraic number theory*, third ed., Springer-Verlag, 1996.

[7] H. Cohen, *Advanced topics in computational number theory*, Springer-Verlag, 2000.

[8] David A. Cox, *Primes of the form $x^2 + ny^2$*, A Wiley-Interscience Publication, John Wiley & Sons Inc., New York, 1989, Fermat, class field theory and complex multiplication.

[9] R. Crandall, C. Pomerance, *Prime numbers, a computational perspective*, Springer-Verlag, 2001.

[10] Graham Everest and Thomas Ward, *Heights of polynomials and entropy in algebraic dynamics*, Universitext, Springer-Verlag London Ltd., London, 1999.

[11] Xavier Gourdon, *Algorithmique du théorème fondamental de lalgèbre*, Rapport de recherche 1852, INRIA, 1993.

[12] Peter Henrici, *Applied and computational complex analysis*, Wiley- Interscience [John Wiley & Sons], New York, 1974, Volume 1: Power series integrationconformal mappinglocation of zeros, Pure and Applied Mathematics.

[13] Henryk Iwaniec and Emmanuel Kowalski, *Analytic number theory*, American Mathematical Society Colloquium Publications, vol. 53, American Mathematical Society, Providence, RI, 2004.

[14] Serge Lang, *Algebra*, second ed., Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1984.

[15] Serge Lang, *Algebraic number theory*, second ed., Graduate Texts in Mathematics, vol. 110, Springer-Verlag, New York, 1994.

[16] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovasz, *Factoring polynomials with rational coefficients*, Math. Ann. **261** (1982), no. 4, 515-534.

[17] Wladyslaw Narkiewicz, *Elementary and analytic theory of algebraic numbers*, second ed., Springer-Verlag, Berlin, 1990.

[18] Christos H. Papadimitriou, *Computational complexity*, Addison-Wesley, 1994.

[19] Pierre Samuel, *Théorie algébrique des nombres*, Hermann, Paris, 1967.

[20] J.-P. Serre, *A course in arithmetic*, Springer-Verlag, New York, 1973, Translated from the French, Graduate Texts in Mathematics, No. 7.

[21] J. H. Silverman, *The arithmetic of elliptic curves*, Springer-Verlag, New York, 1986.

[22] A. Storjohann, *Algorithms for matrix canonical forms*, Ph.D. thesis, ETH Zurich, 2000, `http://www.cs.uwaterloo.ca/~astorjoh/dissA4.ps.`

[23] G. Tenenbaum, *Introduction à la théorie analytique et probabiliste des nombres*, Pub. Inst. Elie Cartan, 1990.

[24] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, Cambridge University Press, New York, 1999.