Computation in Algebra and Arithmetic

ETH Zürich, Spring Semester 2022

David Loeffler

June 2, 2022

Contents

1 Introduction

1.1 What this course is (and isn't) about

In this course, we'll learn how to do interesting mathematical calculations on a computer, focussing on examples coming up in algebra, algebraic geometry, and number theory.

This isn't a computer *programming* course; we won't be writing complicated software of our own. Rather, we'll be learning about some of the software other people have already written, which problems it can solve, how to use it, and a slight hint about *how* it does these computations.

Exact computation In this course we're going to focus on questions which have *exact answers*, as questions in algebra generally do. We'll stick to algorithms which are *provably correct*, so if the computer says the answer is $\frac{3}{4}$, then it's a theorem that the answer is really $\frac{3}{4}$ (unless our program has a bug!).

This is a very different world from questions that come up in analysis (e.g. "compute the value at $x = 1$ of the function defined by this differential equation") where often there's no tidy formula for the answer, so the best we can do is to find an approximation (and the main challenge is how to avoid the error caused by inexact approximations from accumulating too fast).

Efficiency We also (usually) won't worry about doing things in the speediest, most efficient manner. There is a huge amount of research which goes into doing basic operations – like multiplying two integers, or solving a system of linear equations – in the most asymptotically efficient way, but that won't be our concern here; we're more interested in finding *any* way of doing computations with complicated abstract objects (like algebraic varieties, for instance).

1.2 Our toolkit

We'll mostly use the following software:

- **Sage**: open-source mathematics software with a focus on algebra and number theory.
- **Maple**: commercial, mostly oriented towards symbolic and real-number computations but with some algebra functionality.

Sage and Maple are available on the `euler.ethz.ch` computational server, which all ETH staff and students can log into.

You can also download and install them on your own computer – Sage is free anyway, while you can get Maple licenses through ETH – or Sage can be accessed through a free cloud service called CoCalc (`https://cocalc.com/`).

There are other more specialised packages such as Magma and GAP which we'll use for particular bits of the course, but those two will cover most things.

1.3 Computing mathematical objects

Let A be some interesting set (e.g. A could be the integers $\mathbb{Z}$, or something more exotic, like the ideal-class group of your favourite number field). What does it mean to represent elements of A on a computer?

1.3.1 Exact data-types

Things like integers, rational numbers, integers mod N, etc are *exact*: there is a way to describe them by a finite amount of data (ultimately, finite strings of ones and zeroes). Of course, this is only possible if A is countable! Then we can ask:

- Are there algorithms which will compute the operations we care about?
- Are there algorithms which will check, given some string of data, whether it represents a valid element of A?
- Are there algorithms which will check, given two data strings, whether they represent the same element of A?

The last one is quite important – often there's no unique "best" representative, so two different data strings might be the same element of A, and it's important that we should be able to test equality. If we have a way of representing A on a computer satisfying these conditions, we say A is an *exact datatype*.

Remark. Not all countable mathematical structures can be represented exactly on a computer, because of problems with equality checking. For instance, the set of all algorithmically computable functions $\mathbb{Z} \to \mathbb{Z}$ is countable, but there's no algorithm which will check whether two given algorithms compute the same function (it's an undecidable problem).

1.3.2 New structures from old

Generally our data-types will be built up from pre-existing ones. E.g. if our computer already knows how to work in $\mathbb{Z}$, we can represent a rational $x \in \mathbb{Q}$ as a pair (a, b) with $b \neq 0$, representing the rational $\frac{a}{b}$.

Similarly, if we can exactly represent some ring A and compute its ring operations, then we can do exactly represent polynomial rings $A[X_1, \ldots, X_n]$, matrices over A, etc., and do basic arithmetic with these too.

1.3.3 Integers

Even the very first example turns out to be quite subtle! Computers have very fast routines built into their processor hardware for computing with "integers" – but they usually only allocate a fixed block of memory for each number, so they can only handle integers up to a limited size (often the limit is 2^{31} or 2^{63}, for a block of 32 or 64 bits with one bit for sign). If you ask it to do some arithmetic operation where the answer is too big to fit, then the computer might raise an error, or silently return nonsense, or crash. This is called an *integer overflow*.

For doing arithmetic in $\mathbb{Z}$ without overflow errors, there are tricks involving dividing up the bits of an integer into a variable number of smaller blocks. This is referred to as *arbitrary-precision arithmetic* or just *bignum arithmetic*. All of the mathematics software we'll be using will handle bignum arithmetic for you, so you don't have to think about it; but if you're writing your own programs in a general-purpose language like Java or C, it's something to watch out for – you might have to import some special library such as `GMP`.

Remark. Conversely, using bignum routines for small integers is much slower than hardware arithmetic; so if you know that some particular variable in your computation will fit in a machine integer datatype, then you can speed things up by using one. But can you *prove* that x will never be bigger than $2^{31} - 1$?

The basic operations on integers are addition, subtraction, multiplication, and two less obvious ones:

- *Division with remainder*: given $a, b \in \mathbb{Z}$ with $b \neq 0$, compute q, r such that $a = qb + r$ with $0 \leqslant r < |b|$.
- *Extended greatest common divisor*: given $a, b \in \mathbb{Z}$, compute the greatest common divisor $c = \gcd(a, b)$, together with λ, μ such that $\lambda a + \mu b = c$.

The λ, μ of the extended GCD problem "come for free" if we compute $\gcd(a, b)$ via Euclid's algorithm, and provide useful extra information.

Much more sophisticated are questions about *prime factorisation*: testing if some $n \in \mathbb{Z}$ is prime, and if not, decomposing it into its prime factors. It's obvious that algorithms for these problems exist, but how to solve them *efficiently* is a much deeper question, and one that many brilliant researchers have spent whole careers thinking about. We'll come back to this later in the course.

Remark. Maple has a function `isprime()`, but it doesn't actually test if n is prime! If the function returns False, then n is (provably) composite; but it could possibly be making errors the other way, mis-identifying some large composite numbers as primes. Sage's `is_prime()`, on the other hand, returns provably correct output (unless you explicitly tell it to cut corners); but it's much slower.

1.3.4 Approximating the reals

The real numbers $\mathbb{R}$ clearly can't be exactly represented, because they're uncountable.

One can consider the ring of *computable real numbers* – real numbers α for which there exists an algorithm which, given N, computes the first N decimal digits of α. This is a countable set (there are only countably many algorithms), and we can represent its elements on a computer (just store the algorithm). The problem is that *equality of computable real functions is an undecidable problem* – there is no algorithm which inspects two algorithms and determines whether they give the same output.

There are various ways around this. For $\mathbb{R}$, most computers offer something called *floating-point arithmetic*, where you fix a precision $P \geqslant 1$, and represent reals via binary expansions with P significant bits; then arithmetic operations are "rounded off", discarding anything beyond the P'th significant bit. However, even if all your arithmetic is done to precision P, it doesn't mean your final answer is correct up to the P'th bit. The rounding errors accumulate with each arithmetic operation: if x rounds to x', and y rounds to y', then $x + y$ doesn't necessarily round to $x' + y'$. So this kind of arithmetic **does not give provable answers**.

There is something called *interval arithmetic* which works around this by storing, alongside each real-number variable, a count of how many bits of precision it is accurate to, so the variable is something like "3.142 to 4 significant figures" (or its binary equivalent), meaning "some real number in the interval $3.1415 \leqslant x \leqslant 3.14125$".

At each arithmetic operation, interval-arithmetic programs will find a box which (provably!) encloses the answer. E.g. if "$x = 3.142$ to 4 significant figures", then x^2 is in the interval

$$3.1415^2 \leqslant x \leqslant 3.1425^2 \qquad i.e. \qquad 9.86902225 \leqslant x^2 \leqslant 9.87530625,$$

so we can (provably) represent x^2 as "9.9 to 2 significant figures". Notice we lost quite a lot of accuracy in just one step – we can't be sure whether x is 9.87 or 9.88 to 3 sf. However, at least we have a provable bound for how much accuracy is lost.

The problem is that interval arithmetic can never prove that two things are exactly equal. If you are calculating some quantity x and you suspect that $x > 0$, then you have a fighting chance of proving it with interval arithmetic. However, if you suspect that $x = 0$, then interval arithmetic can't ever prove that; you might show $-1/100 < x < 1/100$, or with more work $-1/1000 < x < 1/1000$, but you'll never show it's zero!

1.4 A subtle example: the algebraic real numbers

1.4.1 Theory

Definition. *A real number $\alpha \in \mathbb{R}$ is* algebraic *if there exists a nonzero $f \in \mathbb{Q}[x]$ with $f(\alpha) = 0$.*

As most of you will have seen, the algebraic numbers are a countable subfield of $\mathbb{R}$. In particular, if α and β are algebraic (and we know squarefree polynomials P, Q having x, y as roots), then we can compute polynomials having $\alpha \pm \beta$, $\alpha\beta$, or α/β as roots, using the theory of symmetric functions.

It turns out we can compute effectively with algebraic reals, using the following key result:

Theorem (Sturm, 1829)**.** *There exists an algorithm which, given $P \in \mathbb{Q}[X]$ and $a < b \in \mathbb{Q} \cup \{\pm\infty\}$, will determine the number of roots of P in $[a, b]$.*

(We won't prove this here – it might be a nice project topic.)

So we can represent an algebraic α by a pair consisting of

- a rational polynomial such that $f(\alpha) = 0$,
- an interval $[a, b]$ containing α **and no other root of** f (an "isolating interval").

If we want to add two algebraics $\alpha_1 = (f_1, a_1, b_1)$ and $\alpha_2 = (f_2, a_2, b_2)$, then we compute a polynomial which kills $\alpha_1 + \alpha_2$, and we check if it has a unique root in $[a_1 + a_2, b_1 + b_2]$. If so, we're good. If not, then we subdivide the intervals $[a_1, b_1]$ and $[a_2, b_2]$ into smaller intervals, check which of them contain α_1 and α_2 (using the Theorem), and try again. Eventually our interval will be small enough to isolate $\alpha_1 + \alpha_2$.

Exercise. Given two algebraic numbers $\alpha_i = (f_i, a_i, b_i)$, how might one check if $\alpha_1 = \alpha_2$?

Once we have the field $\mathbb{R}^{\text{alg}}$ of algebraic reals, we can get the field $\mathbb{C}^{\text{alg}}$ of algebraic complex numbers, since $x \in \mathbb{C}$ is algebraic iff its real and imaginary components are (that is, $\mathbb{C}^{\text{alg}} = \mathbb{R}^{\text{alg}} + i\mathbb{R}^{\text{alg}}$).

Remark. Note the similarity to interval arithmetic, but also the differences: because we also store the polynomial that α satisfies, an algebraic number represented in this form is an **exact object**.

1.4.2 Implementations

All this is implemented in Sage (it uses the names `AA` and `QQbar` for $\mathbb{R}^{\text{alg}}$ and $\mathbb{C}^{\text{alg}}$).

```
sage: R.<x> = QQ[]      # set up polynomial ring over Q
sage: f = x^5 - 17*x - 1
sage: a,b,c = f.roots(AA, multiplicities=False) # f has exactly 3 real roots
sage: a
-2.015563046951700?
```

Note the question-mark `-2.015563046951700?`. That is how Sage outputs interval-arithmetic variables – the idea is that the decimal places *before* the question-mark are completely known, while the last one might be off by ±1, so Sage is telling us that

$$-2.015563046951701 \leqslant a \leqslant -2.015563046951699$$

which is more than enough accuracy to isolate the root. However, Sage also stores the polynomial, so it can refine the interval further if we ask it to:

```
sage: a._descr          # internal representation of a
a where a^5 - 17*a - 1 = 0 and a in -2.015563046951700?
sage: a.n(prec=100)     # numeric evaluation to higher precision
-2.0155630469517000366964260152

sage: d = a + b
sage: d.n(100)
```

```
-2.0743866177928034427084350091

sage: g = x^10 + 51*x^6 + 11*x^5 - 1156*x^2 + 68*x - 1
sage: g(d) == 0    # test equality (with proof)
True
```

Maple doesn't have a full implementation of arithmetic in $\mathbb{R}^{\mathrm{alg}}$, but it does have a command `RootFinding[Isolate]` which will compute isolating intervals for each real root of a polynomial:

```
> with(RootFinding):
> Isolate(x^5 - 17*x - 1, output='interval');
      -9518227377035325282905  -4759113688517662641439
[x = [-----------------------, -----------------------],
      4722366482869645213696   2361183241434822606848

         -542551678403779127  -135637919600944775
    x = [-------------------, -------------------],
         9223372036854775808  2305843009213693952

         143903396403431  71951698201729
    x = [---------------, --------------]]
         70368744177664   35184372088832
```

(Note that the denominators are large powers of 2.)

2 Linear Algebra

2.1 Linear algebra over fields

2.1.1 Echelon form

Let F be some field. We suppose elements of F are exactly representable on a computer, and we have algorithms for the field operations (addition, multiplication, inversion) – think $F = \mathbb{Q}$, or $F = \mathbb{F}_p$ for some prime p if you prefer. Then, as we've seen, we can do basic arithmetic with matrices (adding and multiplying).

Remark. If you learn one thing from this course, learn this: computers can do matrix arithmetic over exact fields, and they are *very good at it*. I can't remember the last time I did a matrix multiplication larger than 3×3 by hand.

In practice, we care about matrices as a way of representing linear maps, so we want to consider the effect of changing bases.

Definition. *A matrix (over any commutative ring A) is in* row echelon form *if:*

- *any zero rows come below all the non-zero rows;*
- *for each $i > 1$, the first non-zero entry in the i-th row (if any) is* ***strictly to the right*** *of the first non-zero entry in the $(i-1)$-st row.*

We call these first non-zero entries ***pivots****. It is in* reduced row echelon form *(RREF) if:*

- *the pivot entries are all 1;*
- *the other entries in the same column as a pivot are all 0.*

Theorem (Gaussian elimination)**.** *For any $m \times n$ matrix[1] M over a field F, there is a unique matrix E such that:*

- *E is in RREF,*
- *there exists a non-singular U (not necessarily unique) such that $E = UM$.*

Moreover, given M, we can compute E.

[1] m rows, n columns; and m_{ij} is the entry in the i-th row and j-th column, for $0 \leqslant i \leqslant m$ and $0 \leqslant j \leqslant n$.

I'm sure you've seen this before many times, I just want to emphasise that it's all completely algorithmic. Note that if you want to know the transformation matrix U, then we can get it for free by padding A: if we form the augmented matrix $A' = (A \mid I_m)$ with m rows and $n+m$ columns, then the RREF of A' is $(E \mid U)$, where E is the RREF of A, and U is a nonsingular matrix such that $UA = E$.

Remark. A square matrix M over F is nonsingular iff its RREF is the identity matrix; so by computing the transformation putting M in RREF, we've found the inverse of M.

Remark. Computing echelon form of an $n \times n$ square matrix can be done in $O(n^3)$ field operations. As a by-product we can compute $\det(A)$ at the same time; this is *vastly* quicker than computing it using the definition of the determinant as a sum over permutations, which needs roughly $n!$ steps.

2.1.2 Subspaces

Definition. *Let $V = F^n$, and let $W \leqslant V$ be a subspace. A* reduced echelon basis *of W is an ordered set of non-zero vectors $b_1, \dots, b_m$ forming a basis of W, such that the $m \times n$ matrix with i-th row b_i is in RREF.*

Proposition. *Each subspace $W \leqslant F^n$ has a* ***unique*** *reduced echelon basis. Moreover, given any set of vectors $w_1, \dots, w_r$, we can compute the reduced echelon basis $b_1, \dots, b_m$ of the subspace W spanned by the w_i, and express each w_i as a linear combination of the b_i or vice versa.*

Proof. This is a reformulation of Gaussian elimination, because two matrices A, B satisfy $A = UB$ for some invertible U iff the rows of A and the rows of B span the same subspace. □

MORAL: Subspaces of F^n are a datatype, and the echelon basis gives a unique "best" description – a *normal form* – for a subspace.

We get, essentially for free, algorithms to do the following:

- Compute the sum $W_1 + W_2$ of two subspaces of F^n. (Just stack their basis matrices, one on top of the other, and compute the RREF of the result.)
- Compute whether one subspace is contained in another. (Special case of the above, since $W_1 \subseteq W_2 \Leftrightarrow W_1 + W_2 = W_2$.)
- Check whether some given $v \in F^n$ lies in W, and if so, express it in terms of the basis. (Compute the echelon form of the enlarged matrix given by putting a copy of v along the bottom of the basis matrix of W.)

2.1.3 Kernels and images

If we think of a matrix M as representing a linear map, we might want to compute its kernel and image. It's easiest to do this if we think of our linear maps acting "from the right", i.e. M represents the map

$$A : F^m \to F^n, \qquad A(v) = vM.$$

Then we might want to compute its kernel and image.

Proposition. *Consider the $m \times (n+m)$ matrix $(M \mid I_m)$. Then the RREF echelon form of this matrix has the form*

$$\left(\begin{array}{c|c} \text{\textit{REB of} im } A & \textit{junk} \\ \hline 0 & \text{\textit{REB of} ker } A \end{array}\right)$$

Proof. If the echelon form is $(E \mid C)$, then by construction E is the echelon form of M, so its nonzero rows are the REB of the image of M, and the number of zero rows at the end is $m - \dim \operatorname{im} A = \dim \ker A$.

Since $CM = E$, A sends each row of C to the corresponding row of E, so the last p rows of C are in $\ker A$. These rows are already in RREF, and none of them are zero because C is invertible. By the rank-nullity theorem they must be the REB of $\ker A$. □

Remark. The junk isn't really junk: it's telling us a preimage in F^m of each vector in the echelon basis of $\operatorname{im}(A) \subset F^n$.

So the RREF basis is really the "Swiss army knife" of matrix algebra (over fields): once we have this one tool, we can solve essentially all the problems we want by making straightforward transformations to our input data before feeding it into the algorithm.

Exercise.

(a) Given echelon-form bases for two subspaces W_1, W_2, how would you compute the echelon basis for the intersection $W_1 \cap W_2$?

(b) Given a subspace $W \subseteq F^m$, and an $m \times n$ matrix defining a linear map $A : F^m \to F^n$, how would you compute the subspace $A^{-1}(W)$?

2.1.4 Eigenspaces

If we're thinking of an $n \times n$ matrix M as an endomorphism – a map from F^n to itself – then we might want to know about its eigenvalues and eigenvectors, and hence compute its Jordan normal form.

We can compute the characteristic polynomial $\det(XI_n - M)$, since we can apply determinant algorithms over the field $F(X)$. Assuming we can find the roots of this polynomial (and that they lie in F), then we can use our kernel-finding algorithms to compute, for each root, the eigenspace $\ker(\lambda I_n - M)$, and more generally the rank m generalised eigenspace $\ker\left[(\lambda I_n - M)^m\right]$ for each $m \geqslant 1$. From here, it is an easy step to computing Jordan form.

2.2 Linear algebra over $\mathbb{Z}$

2.2.1 Hermite form

Definition. *A matrix over $\mathbb{Z}$ is in **row Hermite form** if it is in row echelon form, and:*

- *the pivots are positive integers;*
- *the entries above each pivot are in the range $[0, P)$ where P is the corresponding pivot.*

This is a little bit less tidy than RREF over fields, but it's not too far off.

Theorem (Hermite). *Let M be an integer matrix. Then there exists a unique matrix H such that*

- *H is in row Hermite form,*
- *we have $H = UM$ for some invertible[2] integer matrix U.*

Moreover, given M, we can compute H.

We'll need a lemma first:

Lemma. *Let $x_1, \ldots, x_n \in \mathbb{Z}$, and let $z = \gcd(x_1, \ldots, x_n)$. Then we can compute an $n \times n$ invertible integer matrix U such that $U \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} z \\ 0 \\ \vdots \\ 0 \end{pmatrix}$.*

Proof. We give the proof for $n = 2$ (the general case follows easily from this by induction).

If $x_1 = x_2 = 0$ then we can take $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ to be the identity, so assume this isn't the case.

Do an extended GCD to find λ, μ such that $\lambda x_1 + \mu x_2 = z$. This implies λ, μ must be coprime[3], so we can find λ', μ' such that $\lambda\lambda' + \mu\mu' = 1$. Thus $\begin{pmatrix} \lambda & \mu \\ -\mu' & \lambda' \end{pmatrix}$ is invertible; and it maps $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ to $\begin{pmatrix} z \\ w \end{pmatrix}$ for some w. But w is a linear combination of x_1 and x_2, so it must be divisible by z, say $w = hz$ for some h. Thus $\begin{pmatrix} \lambda & \mu \\ -\mu' - h\lambda & \lambda' - h\mu \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} z \\ 0 \end{pmatrix}$. □

Proof of Theorem: existence. The proof is rather similar to Gaussian elimination. We'll prove by induction the following statement:

Claim. For any $0 \leqslant k \leqslant n$, there exists an invertible $m \times m$ matrix U_k such that the first k columns of $U_k M$ are in Hermite form.

[2] This means it has an inverse *in the ring of integer matrices*, so its determinant has to be ±1, not just nonzero.

[3] Note this step would fail if $x = y = 0$!

Clearly $k = 0$ is vacuously true. So assume $k \geqslant 1$ and the statement holds for $k - 1$. WLOG we can assume M itself has the first $k - 1$ columns in Hermite form, and write M in the form

$$\left(\begin{array}{c|c} E & B \end{array} \right) \qquad or \qquad \left(\begin{array}{c|c} E & B \\ \hline 0 & C \end{array} \right)$$

where E is in Hermite form with all its rows nonzero; let's say E has h rows.

If C isn't there (i.e. $h = m$), or the first column of C is zero, then M is already in Hermite form up to column k; so we are done.

Otherwise, let $= \begin{pmatrix} c_1 \\ \vdots \\ c_{m-h} \end{pmatrix}$ be the first column of C. Let U' be an $(m - h) \times (m - h)$ invertible matrix such that $U \cdot = \begin{pmatrix} P \\ 0 \\ \vdots \\ 0 \end{pmatrix}$, where $P = \gcd() \geqslant 1$; and let $U = \left(\begin{array}{c|c} I_h & 0 \\ \hline 0 & U \end{array} \right)$. Then we have

$$U \cdot M = \left(\begin{array}{c|c|c} & a_{1k} & \\ E & \vdots & \star \\ & a_{hk} & \\ \hline & P & \\ & 0 & \\ 0 & \vdots & \star \\ & 0 & \end{array} \right)$$

This is almost in Hermite form up to column k, but the a_{ik}'s might not be in $[0, P)$. However, if we left-multipy by a matrix of the form

$$\begin{pmatrix} 1 & & & t_1 & \\ & \ddots & & \vdots & \\ & & 1 & t_h & \\ & & & 1 & \\ & & & & \ddots \end{pmatrix},$$

with the t's in the $(h + 1)$'st column, this replaces a_{ik} with $a'_{ik} = a_{ik} + Pt_i$; using division-with-remainder, we can choose t_i such that a'_{ik} is reduced modulo the pivot. This gives a new matrix in which the first k columns are Hermite-form, proving the claim. Setting $k = n$, the theorem is proved. □

Remark. We won't prove the uniqueness. It's a bit tedious, and it's also very specific to $\mathbb{Z}$, while existence is more general: it shows that row-echelon form exists, and is computable, over any ring in which we have an extended GCD algorithm (i.e. anything that is "computably a PID"). For uniqueness, we need to be able to choose a preferred generator of each ideal of A, and a preferred generator of each residue class modulo each ideal.

Corollary. *Let G be a subgroup of the abelian group $\mathbb{Z}^n$. Then G has a unique ordered generating set $b_1, \ldots, b_m$ such that the matrix with b_i as rows is in Hermite form; and this Hermite-form generating set can be computed effectively starting from any finite set of generators of G.*

So subgroups of $\mathbb{Z}^n$ are a datatype; and, as before, we get algorithms for computing sums of subgroups, checking whether a vector is in a subgroup, etc, all by feeding slightly different inputs into the Hermite-form algorithm.

Remark. Computing the kernel and image of a map $\mathbb{Z}^m \to \mathbb{Z}^n$ works as before, but it's a little more fiddly to justify *why* it works: the argument we gave before just shows that the last p rows of the echelon basis generate a submodule of $\ker(A)$ of the correct rank. Why is it the whole of $\ker(A)$?

2.2.2 Smith normal form

Definition. *A matrix over $\mathbb{Z}$ is in* Smith normal form *if it is diagonal, with the entries along the diagonal non-negative, and $d_{ii} \mid d_{i+1,i+1}$ for each i.*

Proposition. *For any M there is a unique S such that S is in Smith normal form and $S = UMV$ for invertible U, V; and we can compute S and (some choice of) U, V effectively given M. The diagonal entries of S are called the* elementary divisors *of M.*

The proof-construction is rather reminiscent of Hermite form, although more complicated.

Generally, while Hermite form is useful for questions about specific subgroups, Smith form is useful for asking questions about subgroups *up to isomorphism*. The classic one is the following:

Proposition. *Let G be an abelian group with finitely many generators $g_1, \ldots, g_n$ and finitely many relations*

$$a_{11}g_1 + \cdots + a_{1n}g_n = 0, \quad \ldots, \quad a_{m1}g_1 + \cdots + a_{mn}g_n = 0.$$

Then G is isomorphic to

$$(\mathbb{Z}/d_1) \times \cdots \times (\mathbb{Z}/d_r) \times \mathbb{Z}^{n-r},$$

where $d_1, \ldots, d_r$ are the non-zero elementary divisors of the matrix $A = (a_{ij})$.

This is, of course, exactly the classification of finitely-generated abelian groups; but the point is that the classification is *effectively computable*. Moreover, we can express the generators for the cyclic factors of G in terms of the original generators, and write the relations in the new presentation in terms of the old ones, using the transformation matrices U, V relating A to its Smith form. If $S = UAV$ is the Smith form, and $h_1, \ldots, h_n$ are the elements defined by

$$\begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix} = V^{-1} \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix}$$

then the h_i are a generating set in which the only relations are $h_i^{d_i} = 0$ for each i. (The matrix U tells us how to derive the relations for the new basis for the relations for the old basis.)

Remark. This is useful for groups that arise naturally as the quotient of one "big" group by another; for instance, ideal class groups of number fields, or (co)homology groups of simplicial complexes. Using Smith form gives you nice generating sets for these groups.

3 Polynomials in one variable, I

We're now going to talk about *polynomials* (in one variable) over fields, and (a very closely related topic) *extensions of fields*. Throughout this chapter F is a field datatype; we want to avoid non-separable fields, so we'll assume that either F has characteristic 0, or F has finite characteristic p and every element of F has a computable p-th root.

3.1 Generalities

We have *division with remainder* for polynomials over F: given $a, b \in F[X]$ with $b \neq 0$, we can write $a = bq + r$ where $\deg(r) < \deg(b)$. So we also have GCD's, and even XGCD's, via Euclid's algorithm.

Definition. *A non-constant monic polynomial f is* square-free *if there is no non-constant polynomial $g \in F[X]$ such that $g^2 \mid f$.*

Note that f is square-free in $F[X]$ iff its roots in $\overline{F}$ are distinct. We can test for this very easily:

Lemma. *f is square-free if and only if $\gcd(f, f') = 1$, where f' is the formal derivative of f.*

Proof. Let $\alpha \in \overline{F}$ be a root of f. Then $f(X) = (X - \alpha)^r h(X)$ for some $r \geqslant 1$ and h with $h(\alpha) \neq 0$. Thus $f'(X) = r(X - \alpha)^{r-1} h + (X - \alpha)^r h'$. Hence $\operatorname{ord}_\alpha(f') = r - 1$, unless $r = 0$ in F, in which case it is r; in either case, we have $f'(\alpha) = 0 \Leftrightarrow r > 1$. □

Proposition (Squarefree factorisation)**.** *For each monic $f \in F[X]$, we can compute square-free, pairwise coprime, monic polynomials $f_1, \dots, f_m \in F[X]$ (for some $m \leqslant \deg(d)$) such that*

$$f = \prod_{i=1}^{m} (f_i)^i.$$

Proof. An easy argument from Galois theory shows that such a factorisation exists, and is unique (because any two roots of f in the same $\operatorname{Gal}(\overline{F}/F)$-orbit have the same multiplicity); the problem is to find it.

We start by computing the formal derivative f', and we set

$$a_1 = f / \gcd(f, f'), \qquad b_1 = \gcd(f, f'), \qquad c_1 = a_1 / \gcd(a_1, b_1).$$

If F has characteristic 0, then we have

$$\gcd(f, f') = \prod_{i \geqslant 2} f_i^{(i-1)}$$

(up to scaling by $F^\times$); thus $a_1 = \prod_{i \geqslant 1} f_i$ and hence $c_1 = f_1$. We now iterate by setting

$$a_{i+1} = a_i / c_i, \qquad b_{i+1} = b_i / a_{i+1}, \qquad c_{i+1} = a_{i+1} / \gcd(a_{i+1}, b_{i+1}),$$

and keep going until $a_i = 1$; one sees by induction on i that $a_i = \prod_{j \geqslant i} f_i$ and $b_i = \prod_{j > i} f_i^{(j-i)}$ for all i, so $c_i = f_i$.

If F has characteristic $p > 0$ the situation is a little more subtle: we have

$$\gcd(f, f') = \prod_{p \nmid i} f_i^{(i-1)} \cdot \prod_{p \mid i} f_i^i$$

and so this method only detects the f_i for $p \nmid i$ (we have $c_i = 1$ for $p \mid i$). However, we can keep going until $a_n = 1$, in which case b_n is *purely inseparable*, i.e. has the form $g(X^p)$ for some $g \in F[X]$. Then we can write

$$g(X^p) = \sum a_i X^{pi} = (h(X))^p, \qquad h(X) = \sum_i a_i^{1/p} X^i$$

and apply the above algorithm to h (which has degree $\leqslant \frac{1}{p} \deg(f)$, so the process must terminate). □

3.2 Finite fields

The simplest fields are, of course, finite fields, so we'll start with those.

3.2.1 Setup

Here are some basic properties of finite fields.

- For every prime power $q = p^k$, there is a unique (up to isomorphism) finite field $\mathbb{F}_q$ with q elements, and every finite field is isomorphic to one of these.
- The characteristic of $\mathbb{F}_{p^k}$ is p.
- Every element of $\mathbb{F}_q$ satisfies $x^q = x$. In particular, if $q = p^k$, then $x = (x^{p^{k-1}})^p$, so p-th roots exist and are computable.
- $\mathbb{F}_{q'}$ contains a subfield isomorphic to $\mathbb{F}_q$ if and only if $q' = q^r$ for some r, and this subfield is uniquely determined (it is exactly the elements of $\mathbb{F}_{q'}$ satisfying $x^q = x$).
- For any r, $\mathbb{F}_{q^r}$ is a Galois extension of $\mathbb{F}_q$ and its Galois group is the cyclic group of order r generated by the *Frobenius automorphism* $x \mapsto x^q$.
- The unit group $\mathbb{F}_q^\times$ is a cyclic group of order $q - 1$.

If these properties aren't familiar to you, then you might want to consult a book on Galois theory. (I like Ian Stewart's book, but there are many others.)

Proposition. *For any prime power q and $r \geqslant 1$, there exist irreducible monic polynomials $f \in \mathbb{F}_q$ of degree r; and the quotient ring $\mathbb{F}_q[X]/(f(X))$ is isomorphic to $\mathbb{F}_{q^r}$.*

That is, if we know how to represent $\mathbb{F}_q$ (e.g. if q is prime), then we can represent $\mathbb{F}_{q^r}$ as polynomials in X over $\mathbb{F}_q$ of degree $< r$, with addition defined in the obvious way, and multiplication defined modulo $f(X)$.

Example. The polynomial $x^2 + x + 1$ is an irreducible quadratic polynomial over $\mathbb{F}_2$ – in fact it's the only one – so we get a model of $\mathbb{F}_4$ as linear polynomials over $\mathbb{F}_2$.

3.2.2 Factorisation in $\mathbb{F}_q[X]$

We know that $\mathbb{F}_q[X]$ is a *unique factorisation domain* – any polynomial can be written (uniquely) as a product of powers of irreducible polynomials. How can we compute these?

It's clear that this is computable *in principle*. Given f, if it's not irreducible then it has a factor g with $\deg(g) \leqslant \frac{1}{2}\deg(f)$, and there are only finitely many such g so we just try them all. However, if q or $\deg(f)$ is large, this is prohibitively slow – ideally we'd like algorithms which are **polynomial in the size of the input data**, i.e. in $\deg(f)\log(q)$.

Step 1: Square-free factorisation (using the general algorithm above). This reduces us to factorising square-free polynomials.

Step 2: Distinct-degree factorisation. Given a square-free g of degree d, we're going to write

$$g = \prod_{j=1}^{d} g_j$$

where each g_j is a product of (distinct) irreducible polynomials of degree exactly j. The basic input is:

Lemma. *For any N, the product of all monic irreducible polynomials in $\mathbb{F}_q[X]$ of degree dividing N is $X^{q^N} - X$.*

Proof. We have

$$X^{q^N} - X = \prod_{\alpha \in \mathbb{F}_{q^N}} (X - \alpha).$$

So for an irreducible f, we have the equivalences

(f has degree dividing N)

$\iff$ (f splits as a product of distinct linear factors in $\mathbb{F}_{q^N}$)

$\iff$ (f divides $X^{q^N} - X$).

But $X^{q^N} - X$ has distinct roots in $\mathbb{F}_{q^N}$, so it can't be divisible by the square of any polynomial. □

Hence we can compute, for each N, the product

$$\gcd(X^{q^N} - X, g) = \prod_{j|N} g_i,$$

and hence determine the g_j's.

Remark. It's not totally obvious that this step has polynomial running time, because we need to take GCD's of polynomials of very large degree, such as $X^{q^N} - X$. However, it is sufficient for Euclid's algorithm to compute $X^{q^N} \bmod g$. To do this, we use **exponentiation by squaring**: if $q^N = a_0 + 2a_1 + \cdots + 2^h a_h$ is the binary expansion of q^N, then

$$(X^{q^N} \bmod g) = (X \bmod g)^{a_0} \cdot (X^2 \bmod g)^{a_1} \cdot (X^4 \bmod g)^{a_2} \dots,$$

and we can compute $X^2 \bmod g$, $X^4 \bmod g$, etc by squaring and then reducing mod g at each step, so we never go above $2 \deg g$ (and similarly for the steps to put together $X^{q^N} \bmod g$).

Step 3: Equal-degree factorisation. We're now left with the problem of factorising a polynomial h under the assumption that h is square-free and all its irreducible factors have the same degree. This step is rather harder than the first two steps, but good algorithms do exist, such as Berlekamp's algorithm. This would be a good project topic.

3.2.3 Irreducibility

If we just care about testing whether f is irreducible, then steps 1 and 2 are enough, or we can make the following short-cut:

Proposition (Rabin's irreducibility test). *If f has degree N (not necessarily square-free), then f is irreducible iff f divides $X^{q^N} - X$, but for every prime divisor ℓ of N, we have* $\gcd(X^{q^{N/\ell}} - X, f) = 1$.

Proof. If h is an irreducible factor of f of degree M, then either M divides N, or it doesn't. If $M \mid N$ (but $M \neq N$), then $M \mid \frac{N}{\ell}$ for some ℓ, and in this case h is a common factor of f and $X^{q^{N/\ell}} - X$. If $M \nmid N$, then h, and hence f, doesn't divide $X^{q^N} - X$. □

Again, this can be implemented very efficiently using binary exponentiation.

3.2.4 Conway polynomials

(If there had been time, I would have said something at this point about **Conway polynomials**, which give us a way of choosing a "simplest" irreducible polynomial of degree r over $\mathbb{F}_p$ for each r, allowing us to work with the algebraic closure $\overline{\mathbb{F}}_p$. This would be a good project topic.)

3.3 Polynomials over $\mathbb{Q}$ and $\mathbb{Z}$

3.3.1 Preliminaries

There's not a lot of difference between factorization theory over $\mathbb{Q}$ and over $\mathbb{Z}$, because:

Lemma (Gauss). *Let f, g be monic polynomials in $\mathbb{Q}[X]$ with $g \mid f$. If $f \in \mathbb{Z}[X]$, then $g \in \mathbb{Z}[X]$.* □

We can make any polynomial integral by scaling: for a monic $f \in \mathbb{Q}[X]$, we can form the polynomial

$$f_B(X) = B^{\deg f} f(\tfrac{X}{B})$$

for some $B \geqslant 1$. If we choose B sufficiently divisible, then $f_B \in \mathbb{Z}[X]$. Of course, any factor of f goes to a factor of f_B. So if we want to understand factorisation in $\mathbb{Q}[X]$, it's enough to understand factorisation of monic polynomials in $\mathbb{Z}[X]$.

Definition. *If $f = \sum a_n X^n \in \mathbb{Z}[X]$, the* reduction *of f modulo p is the polynomial $\bar{f} = \sum \bar{a}_n X^n$ where $\bar{a}_n$ is the reduction of a_n.*

Proposition. *If f is monic and $\bar{f}$ is irreducible (or square-free), then f is irreducible (resp. square-free).* □

Going the other way doesn't work: things become "more reducible" mod p. For square-freeness this doesn't happen very often:

Proposition. *Let $f \in \mathbb{Z}[X]$ be square-free. Then there is an explicitly computable finite set of primes S such that if $p \notin S$, then $\bar{f} = f \bmod p$ is square-free in $\mathbb{F}_p[X]$.*

Proof. Consider the polynomials $f(X)$ and $f'(X)$. Since f and f' are coprime (in $\mathbb{Q}[X]$), we can use Euclid's algorithm to write

$$af + bf' = 1$$

for some $a, b \in \mathbb{Q}[X]$. Clearing denominators, we can write

$$Af + Bf' = N$$

for some $N \geqslant 1$ and $A, B \in \mathbb{Z}[X]$.

Now let S be the set of primes dividing N. If $p \notin S$, then mod p we can write

$$\bar{A}\bar{f} + \bar{B}\bar{f}' = \bar{N}, \qquad \bar{N} \neq 0.$$

So $\bar{f}$ and $\bar{f}'$ are coprime in $\mathbb{F}_p[X]$. □

Irreducibility is much more subtle: there exist irreducible monic polynomials $f \in \mathbb{Z}[X]$ such that $\bar{f}$ is reducible for *every* prime p (e.g. $X^4 + 1$ is an example). We'll see more on this later.

3.3.2 A bad factorization algorithm

Recall that we can compute in the field $\mathbb{C}^{\text{alg}}$, which is an explicit model for the algebraic closure of $\mathbb{Q}$. In particular, given any polynomial $f \in \mathbb{Q}[X]$, we can compute isolating intervals for its real roots, and isolating boxes for its complex roots. Moreover, given an element $\alpha \in \mathbb{C}^{\text{alg}}$ we can compute whether it lies in $\mathbb{Z}$ or not: just refine the bounding box for α until it contains at most one integer, and check if α is equal to that integer.

Proposition. *Given a monic, squarefree $f \in \mathbb{Z}[X]$ following algorithm computes the factorization of f in $\mathbb{Z}[X]$:*

- *Make a list of the roots of f in $\mathbb{C}^{\text{alg}}$.*
- *For each (nonempty, proper) subset S of the roots, compute $\prod_{\alpha \in S}(X - \alpha)$, and check if its coefficients are integral.*

This shows that factorization in $\mathbb{Q}$ is possible, but it is a woefully bad algorithm in practice. Rather surprisingly, the solution will be to replace approximate real-number information with approximate p-adic information, for a prime p!

4 Polynomials in one variable, II

4.1 Mignotte's bound

Let $f \in \mathbb{Z}[X]$ be monic. We want to get some control of the "size" of potential factors – how big their coefficients can be.

Definition. *Let $1 \leqslant p \leqslant \infty$. The L^p norm of $f = \sum a_i X^i$, is*

$$\|f\|_p = \begin{cases} \left(\sum_i |a_i|^p\right)^{1/p} & \text{if } p < \infty \\ \max_i |a_i| & \text{if } p = \infty. \end{cases}$$

Lemma (Landau). *For any $\alpha \in \mathbb{C}$ and any $g \in \mathbb{C}[X]$ we have have*

$$\|(X - \alpha)g\|_2 = \|(\bar{\alpha}X - 1)g\|_2.$$

Proof. We can write $\|g\|^2 = \langle g, g\rangle$ where $\langle f, g\rangle$ is the usual inner product (linear in f and conjugate-linear in g).

Thus

$$\|(X - \alpha)g\|_2^2 = \langle Xg, Xg\rangle - \alpha\langle g, Xg\rangle - \bar{\alpha}\langle Xg, g\rangle + \alpha\bar{\alpha}\langle g, g\rangle.$$

Similarly

$$\|(\bar{\alpha}X - 1)g\|_2^2 = \alpha\bar{\alpha}\langle Xg, Xg\rangle - \bar{\alpha}\langle Xg, g\rangle - \alpha\langle g, Xg\rangle + \langle g, g\rangle.$$

Since $\langle Xg, Xg\rangle = \langle g, g\rangle$, these expressions are the same. □

We'd like to know an upper bound for the L_p norms of factors of f (for some p, it doesn't really matter which!). We'll do this by relating the L_p norms (which are easy to define, but don't respect multiplication) with another quantity which is multipicative.

Definition. *For any monic f, the* Mahler measure *of f is the quantity*

$$M(f) = \prod_{i=1}^{n} \max(1, |\alpha_i|) \in \mathbb{R}$$

where $\alpha_1, \ldots, \alpha_n$ are the roots of f in $\mathbb{C}$.

Clearly $M(f) \geqslant 1$, and $M(fg) = M(f)M(g)$. In particular, if g is a factor of f then $M(g) \leqslant M(f)$.

Proposition (Landau). *We have*

$$M(f) \leqslant \|f\|_2.$$

Proof. We order the roots so $|\alpha_1|, \dots, |\alpha_r| \geqslant 1$ and the rest are < 1. Then, using the Lemma repeatedly, we have

$$\begin{aligned}\|f\|_2 &= \|(X-\alpha_1)\dots(X-\alpha_n)\|_2 \\ &= \|(\bar{\alpha}_1 X - 1)\dots(\bar{\alpha}_r X - 1)(X-\alpha_{r+1})\dots\|_2\end{aligned}$$

If h is this last polynomial, then clearly $\|h\|_2$ is at least the absolute value of the leading coefficient of h, which is exactly $M(f)$. □

On the other hand:

Proposition. *If f has degree d, then*

$$\|f\|_\infty \leqslant \binom{d}{[d/2]} M(f).$$

Proof. If $f = \sum a_i X^i$, then $(-1)^i a_i$ is the sum of all possible products of subsets of i of the roots. Each such product has norm at most $M(f)$, and there are $\binom{d}{i}$ of them. So $|a_i| \leqslant \binom{d}{i} M(f)$. Since $\binom{d}{[d/2]} = \max_i \binom{d}{i}$, we conclude. □

Combining these we have the following:

Theorem (Mignotte's factor bound). *If $f, g \in \mathbb{Z}[X]$ are monic and $g \mid f$, then we have*

$$\|g\|_\infty \leqslant \binom{d}{[d/2]} \|f\|_2,$$

where $d = \deg(g)$.

Proof. We have

$$\begin{aligned}\|g\|_\infty &\leqslant \binom{d}{[d/2]} M(g) \\ &\leqslant \binom{d}{[d/2]} M(f) \quad \text{(as } g \mid f\text{)} \\ &\leqslant \binom{d}{[d/2]} \|f\|_2 \quad \text{(by Landau).}\end{aligned}$$

□

This gives another factorization algorithm – if if f is not irreducible then it has a factor g whose degree and L_∞ norm are bounded, and there are only finitely many of these, so we can just try them all. Sadly, this is hopelessly inefficient, since the search space has size $O(2^{d^2/2})$. However, it really comes into its own when combined with information modulo prime powers.

4.2 Hensel lifting

4.2.1 Hensel's lemma

Lemma (Hensel). *Let f be a monic polynomial over $\mathbb{Z}$ and suppose that we have $\bar{f} = \bar{g}\bar{h} \bmod p$ for some monic $\bar{g}, \bar{h} \in \mathbb{F}_p[X]$ with $\gcd(\bar{g}, \bar{h}) = 1$.*

Then, for any n, there exist uniquely determined monic polynomials $g_n, h_n \in (\mathbb{Z}/p^n)[X]$ such that $f = g_n h_n \bmod p^n$ and $g_n = \bar{g}, h_n = \bar{h} \bmod p$.

Proof. The case where $\bar{g}(X) = X - \bar{\alpha}$ for some $\bar{\alpha} \in \mathbb{F}_p$ is very standard. The proof in the general case is basically the same as this, but uniqueness requires a little care.

We induct on n. There is nothing to show if $n = 1$. Assume the statement holds for n. Let $\tilde{g}_{n+1}, \tilde{h}_{n+1}$ be arbitrary lifts of g_n, h_n to mod p^{n+1}. Then $\tilde{g}_{n+1}\tilde{h}_{n+1} = f + p^n t$ for some $t \in \mathbb{F}_p[X]$.

Since $\bar{f}$ and $\bar{g}$ are coprime, we can choose $\bar{a}, \bar{b} \in \mathbb{F}_p[X]$ with $\bar{a}\bar{g} + \bar{b}\bar{h} = 1$. We consider the polynomials

$$g_{n+1} = \tilde{g}_{n+1} + p^n(-\bar{b}t + \lambda), \qquad h_{n+1} = \tilde{h}_{n+1} + p^n(-\bar{a}t + \mu),$$

for some arbitrary $\lambda, \mu \in \mathbb{F}_p[X]$. Then we have

$$\begin{aligned} g_{n+1}h_{n+1} &= \tilde{g}_{n+1}\tilde{h}_{n+1} - p^n t(\bar{a}\bar{g} + \bar{b}\bar{h}) + p^n(\lambda\bar{h} + \mu\bar{g}) + (\text{stuff divisible by } p^{2n}) \\ &= f + p^n(\lambda\bar{h} + \mu\bar{g}) \bmod p^{n+1}. \end{aligned}$$

So for any choice of λ, μ with $\lambda\bar{h} + \mu\bar{g} = 0$ gives a factorisation of f modulo p^{n+1}, and conversely any factorisation of f lifting (g_n, h_n) has this form.

Since $(\bar{h}, \bar{g})$ are coprime, this is equivalent to $\lambda = \sigma\bar{g}$, $\mu = -\sigma\bar{h}$ for some σ. Thus we have

$$g_{n+1} = \tilde{g}_{n+1} - p^n\bar{b}t + p^n\sigma\bar{g}, \qquad h_{n+1} = \tilde{h}_{n+1} - p^n\bar{a}t + p^n\sigma\bar{h}.$$

There is a unique choice of σ which will make g_{n+1} monic of degree $\deg \bar{g}$, and comparing leading terms, with this σ we have h_{n+1} monic of degree $\deg \bar{h}$ as well. □

Note this proof is totally algorithmic; and by induction we can extend it to any number of factors.

4.2.2 Factorization by Hensel lifting

Given a monic f in $\mathbb{Z}[X]$, and p such that the mod p reduction $\bar{f}$ is square-free, we can factor it modulo p into irreducibles. Using Hensel, we can lift this to a factorisation of f modulo p^n, and all those factors will be irreducible in $(\mathbb{Z}/p^n)[X]$.

By considering all possible subsets of those factors, we can write down all polynomials dividing f modulo p^n. If f has a factor in $\mathbb{Z}[X]$, say g, then g mod p^n must be in our list of mod p^n polynomials. If $p^n > 2B$, where B is Mignotte's factor bound, then each polynomial mod p^n comes from at most one polynomial over $\mathbb{Z}$ of L_∞-norm $\leqslant B$; and we can check each of these possibilities to see if it's a genuine factor of f.

Example. Consider $f(x) = x^6 - x^5 - 4x^2 + 8$. This is square-free mod p for $p \notin \{2, 3, 71\}$. If we take $p = 7$, then

$$f \bmod 7 = (x^3 + 3x^2 + 2x + 2) \cdot (x^3 + 3x^2 + 3x + 4).$$

If f has a nontrivial factor g in $\mathbb{Z}[X]$, that factor must reduce to a factor of f mod 7; so it has to be cubic. By Mignotte's bound with $d = 3$ we have $\|g\|_\infty \leqslant B = \binom{3}{1}\|f\|_2 = 3\sqrt{82} \leqslant 28$. Since $7^3 \geqslant 2B$, it suffices to Hensel-lift mod 7^3.

We find that

$$f \bmod 343 = \left(x^3 - 39x^2 + 115x - 73\right) \cdot \left(x^3 + 38x^2 - 5x - 33\right).$$

Now, -39 mod 343 can't be the reduction of an integer in the range $[-28, 28]$. So this factorisation can't come from a factorisation in $\mathbb{Z}[X]$ and hence f is irreducible.

(Actually f is irreducible mod 5, so it is obviously irreducible in $\mathbb{Z}[X]$, but I wanted to demonstrate the method.)

Remark. This algorithm is clearly at worst exponential in the number of factors of $\bar{f}$ (which is at most $\deg(f)$, but is often much smaller). There is an improvement due to van Hoeij, which uses LLL reduction algorithms to choose good subsets of the factors; this is known to run in polynomial time, and is the algorithm used by modern computer algebra systems.

4.2.3 Galois groups

If we fix an f (of some degree d), and factorise its mod p reduction for varying p, what happens? For each p we can list the degrees of the irreducible factors, which are integers $(d_1, \ldots, d_m)$ (up to re-ordering) with $d_1 + \cdots + d_m = d$.

Theorem.

(a) If $(d_1, \ldots, d_m)$ are the degrees of the factors mod p, for some p such that the reduction is square-free, then there exists an element in the Galois group $\mathrm{Gal}(f/\mathbb{Q})$ *(the* Frobenius element, *well-defined up to conjugacy) which, as a permutation of the roots of f in $\overline{\mathbb{Q}}$, has m disjoint cycles of lengths $d_1, \ldots, d_m$.*

(b) (Chebotarev's density theorem) If $\mathrm{Gal}(f/\mathbb{Q})$ *has an element with cycle lengths $(d_1, \ldots, d_m)$, then there exist infinitely many primes such that the factors of f* mod *p have degrees $d_1, \ldots, d_m$; and the density of such primes is equal to the fraction of elements of G which have that cycle type.*

Part (a) is fairly elementary and can be found in most algebraic number theory texts. Part (b) is a little harder (depending on how you want to define "density"); see e.g. Theorem 7.30 of Narkiewicz's *Elementary and Analytic Theory of Algebraic Numbers*. The important point is that it gives pretty strong information on what $\mathrm{Gal}(f/\mathbb{Q})$ can be.

Remark. From (a) we see that f mod p can only be irreducible if $\mathrm{Gal}(f/\mathbb{Q})$ contains a d-cycle. This explains my example earlier about $X^4 + 1$: the Galois group is the Klein four-group, which is a transitive subgroup of S_4 but doesn't contain a 4-cycle – the only cycle types are $(1, 1, 1, 1)$ and $(2, 2)$. So for every p, the reduction of $X^4 + 1$ is reducible.

Example. Consider $f(x) = x^6 - x^5 - 4x^2 + 8$. Here's a little script which factorises f mod p for all primes up to 10000, and tells us how often each degree sequence occurs.

```
sage: R.<x> = ZZ[]
sage: f = x^6 - x^5 - 4*x^2 + 8
sage: partcounts = { }
sage: for p in prime_range(10000):
....:     fbar = f.change_ring(GF(p))
....:     if not fbar.is_squarefree(): continue
....:     degs = tuple(g.degree() for (g, e) in fbar.factor())
....:     if degs not in partcounts.keys():
....:         partcounts[degs] = 1
....:     else:
....:         partcounts[degs] += 1
....: for P in sorted(partcounts.keys()):
....:     print("%s : %.2f%%" % (P, partcounts[P] * 100 / prime_pi(10000)))
```

The output looks like this:

```
(1, 1, 1, 1, 1, 1) : 0.73%
(1, 1, 2, 2) : 11.80%
(1, 1, 4) : 24.25%
(1, 5) : 20.59%
(2, 2, 2) : 8.87%
(3, 3) : 16.60%
(6,) : 16.92%
```

There are only 16 subgroups of S_6 that act transitively on $\{1, \dots, 6\}$, and only two of those contain elements with all the above cycle-types: S_6 itself, and another smaller group (which is actually isomorphic to S_5, but acting in a weird way, on 6 points rather than 5). We suspect $\mathrm{Gal}(f/\mathbb{Q})$ can't be S_6, because there are lots of cycle-types we haven't seen, like $(4, 2)$ for example.

In fact Sage knows how to compute Galois groups over $\mathbb{Q}$ [how? – project!] and the above method using reduction is one of the tools it uses. So let's count the number of elements of each cycle-type:

```
sage: G = f.change_ring(QQ).galois_group()
....: C = [tuple(reversed(s.cycle_type())) for s in G]
....: for c in sorted(set(C)):
....:     print("%s : %s" % (c, C.count(c)/G.order()))

(1, 1, 1, 1, 1, 1) : 1/120
(1, 1, 2, 2) : 1/8
(1, 1, 4) : 1/4
(1, 5) : 1/5
(2, 2, 2) : 1/12
(3, 3) : 1/6
(6,) : 1/6
```

This matches pretty closely with the percentages above. In particular, *f is irreducible modulo $\frac{1}{6}$ of all primes*.

5 Commutative algebra

5.1 Ideals

Let F be a field (in which we can compute), and $R = F[X_1, \dots, X_n]$ for some n. We're going to be interested in *ideals* $J \trianglelefteq R$. Any such ideal is finitely-generated (Hilbert's basis theorem), so can be written as $\langle f_1, \dots, f_r \rangle$ for some polynomials $f_i \in R$.

We'd like to solve the following basic problems:

- Ideal membership: given $f_1, \dots, f_r$ and g, is $g \in \langle f_1, \dots, f_r \rangle$, and if so, can we compute $q_i \in R$ such that $g = \sum q_i f_i$?
- Equality of ideals: given $f_1, \dots, f_r$ and $g_1, \dots, g_s$, do they generate the same ideal? (Reducible to ideal membership: just check if each g_i is in the ideal of the f's, and vice versa.)
- Preimage of an ideal: given a map of F-algebras $\phi : F[X_1, \dots, X_n] \to F[Y_1, \dots, Y_m]$ (specified by $X_i \mapsto g_i(Y_1, \dots, Y_m)$ for some polynomials g_i), and an ideal I of $F[Y_1, \dots, Y_m]$, can we compute generators of $\phi^{-1}(I)$?

5.2 The language of algebraic geometry

Why do we care about such problems? Mostly because there's a deep connection between commutative algebra and geometry. It's the geometry which provides the intuition and motivation, while the nitty-gritty computations are on the algebra side.

Definition. *Let $\overline{F}$ be an algebraic closure of F.*

- *Given an ideal $J \trianglelefteq R$, we define $V(J) = \{x \in \overline{F}^n : f(x) = 0 \;\forall\, f \in J\} \subseteq \overline{F}^n$.*
- *Given a subset $S \subseteq \overline{F}^n$, we define $I(S) = \{f \in R : f(x) = 0 \;\forall\, x \in S\}$, which is an ideal of R.*

Theorem (Hilbert's Nullstellensatz)**.** *We have $I(V(J)) = \operatorname{rad}(J)$, where $\operatorname{rad}(J) = \{f : f^r \in J \text{ for some } r \geqslant 1\}$.*

It follows that the maps I and V give a bijection between *radical ideals* (ideals $J \trianglelefteq R$ such that $\operatorname{rad} J = J$) and *F-algebraic sets* (subsets of $\overline{F}^n$ of the form $V(J)$ for some J).

Remark. Note $\{(x, y) \in \mathbb{C}^2 : x = iy\}$ is $\mathbb{C}$-algebraic, but not $\mathbb{Q}$-algebraic. If $F = \mathbb{Q}$ and S is this set, what is $V(I(S))$?

Lemma. *If we can solve ideal membership, then we can also solve the following: given $f_1, \dots, f_r$ and g, test whether $g \in \operatorname{rad} J$, where $J = \langle f_1, \dots, f_r \rangle$.*

Proof. Consider the ring $\widetilde{R} = F[X_1, \dots, X_n, t]$ where t is an extra variable, and let K be the ideal of R generated by J and $tg - 1$.

Then $V(K)$ consists of the points $(x, \frac{1}{g(x)})$ for all $x \in V(J)$ with $f(x) \neq 0$. So it is empty if, and only if, $g = 0$ on $V(J)$; that is, if $g \in \operatorname{rad} J$. So $g \in \operatorname{rad} J \Leftrightarrow 1 \in \operatorname{rad} K \Leftrightarrow 1 \in K$. □

With this in hand, we can test whether $\operatorname{rad} J = \operatorname{rad} J'$, given generating sets of J and J': just test whether each generator of J is in $\operatorname{rad} J'$ and vice versa. So we have a way of testing equality between algebraic sets.

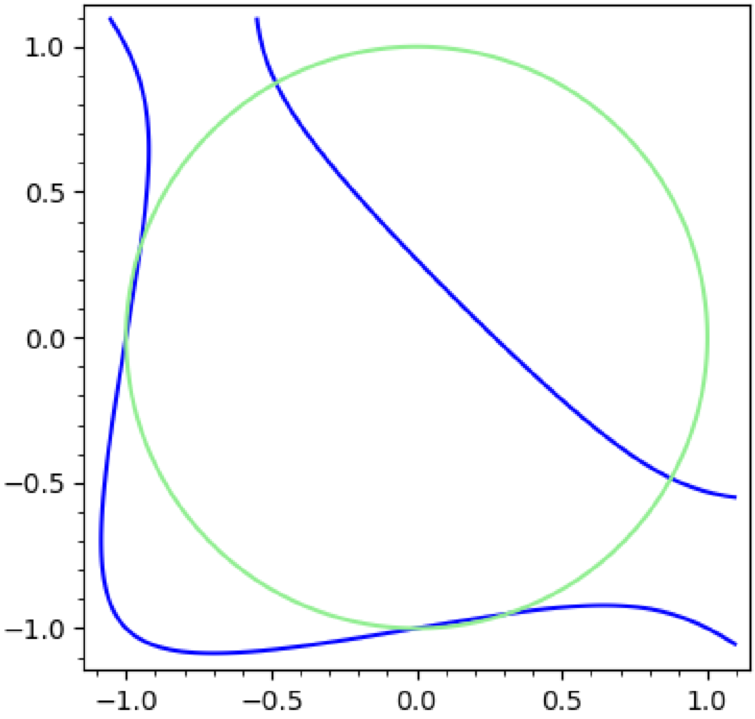
Remark. Note that this isn't the same as being able to compute a generating set of $\operatorname{rad} J$, which is a bit harder.

What geometric problem does "ideal preimage" solve? It corresponds[1] to taking the *image* of an algebraic set under the map $\overline{F}^m \to \overline{F}^n$ sending y to $x = (g_1(y), \dots, g_n(y))$.

Example. Consider the following cubic polynomial in 3 variables:

$$\mathcal{F}(X,Y) : X(X+Y)(X+1) + Y(Y+1)(Y+X) + (X+1)(Y+1) = 4(X+Y)(Y+1)(X+1).$$

I want to compute the intersection points of the curve $\mathcal{C} : \{\mathcal{F}(X,Y) = 0\}$ with the unit circle $X^2 + Y^2 = 1$.



[1] Almost. In general the image of an algebraic set isn't an algebraic set; e.g. mapping $\{x, y : xy = 1\}$ via $(x, y) \mapsto x$, we get $\overline{F} - \{0\}$, which isn't an algebraic set. But ideal preimage computes the smallest algebraic set *containing* the image.

The intersection points are exactly the algebraic set defined by the ideal $I = \langle \mathcal{F}, \mathcal{G} \rangle$. To compute their x-coordinates, we want to find $\phi^{-1}(I)$ where $\phi : \mathbb{Q}[X] \hookrightarrow \mathbb{Q}[X, Y]$ is the obvious inclusion.

5.3 Dividing polynomials

We understand divisibility and ideals for one-variable polynomials very well, because we have *division with remainder* for polynomials (and hence Euclid's algorithm, etc).

We're going to define a sort of long-division algorithm for polynomials in several variables. In the one-variable case, we deal with the highest-degree terms first, and then work downwards. For multiple variables, we need to decide which terms to attack first.

Definition. *A* monomial *is an element $X_1^{a_1} \dots X_n^{a_n} \in R$, for integers $a_i \geqslant 0$. Its* multidegree *is $(a_1, \dots, a_n) \in \mathbb{N}^n$.*

An monomial order *is a total order $\preccurlyeq$ on the monomials of R (equivalently, on $\mathbb{N}^n$) with the following properties:*

- *multiplication is respected, so for all monomials M, N, P we have $M \preccurlyeq N \iff MP \preccurlyeq NP$;*
- *$\preccurlyeq$ is a well-ordering, so any nonempty set of monomials has a least element.*

Remark. One can check that the second condition is equivalent to assuming that $1 \preccurlyeq M$ for every M.

Proposition. *The standard lexicographic order, in which $X^{(a_1,\dots,a_n)} \preccurlyeq X^{(b_1,\dots,b_n)}$ if $a_1 < b_1$, or if $a_1 = b_1$ and $a_2 < b_2$, or (etc), is a monomial order.*

(Note that the sequence goes

$$1 \preccurlyeq X_n \preccurlyeq X_n^2 \preccurlyeq X_n^3 \preccurlyeq \dots \preccurlyeq X_{n-1} \preccurlyeq X_{n-1}X_n \preccurlyeq X_{n-1}X_n^2 \preccurlyeq \dots$$

so any polynomial in X_n alone is "smaller" than any polynomial involving $X_1, \dots, X_{n-1}$.)

Proof. Exercise. □

There are other interesting monomial orders, which have their merits for different problems, but we're going to stick to lex ordering for simplicity. Once we've fixed a monomial ordering, every polynomial has a uniquely-defined *leading term* $\mathrm{LT}(f)$ (which is a nonzero multiple of a monomial), and a *multidegree* $\operatorname{mdeg} f \in \mathbb{N}^d$, which is the vector of exponents in $\mathrm{LT}(f)$.

Theorem (Division algorithm in $F[X_1, \dots, X_n]$)**.** *Let $g_1, \dots, g_s$ be non-zero polynomials, and fix a monomial order $\preccurlyeq$. Then we can write every $f \in F[X_1, \dots, X_n]$ in the form*

$$f = q_1 g_1 + \dots + q_s g_s + r,$$

where no monomial appearing in r is divisible by any of $\mathrm{LT}(g_1), \dots, \mathrm{LT}(g_s)$, and for each i we have $\operatorname{mdeg}(q_i g_i) \preccurlyeq \operatorname{mdeg} f$.

Proof. Let $f_0 \in R$. If there is no monomial in f divisible by any $\mathrm{LT}(g_j)$, we are done. If not, let v_0 be the largest degree of any "bad" monomial in f_0. We can then consider $f_1 = f - \lambda q g_j$, for some j, and some monomial q and scalar λ, chosen to kill off the degree v_0 term. Now any bad monomials in f_1 have to have smaller degree than v_0. Continuing, we obtain a sequence of polynomials $f_0, f_1, \ldots$, in which the sequence "$v_r =$ largest bad exponent in f_r" is strictly decreasing. Since our monomial ordering is a well-ordering, such a sequence must terminate after finitely many steps. □

Note this is a completely computable process.

Remark. It is important to note that the "remainder" r is *not* uniquely determined by the conditions. If our computation gives $r = 0$, then $f \in \langle g_1, \ldots, g_s \rangle$; but it's not remotely obvious that if $r \neq 0$ we have $f \notin \langle g_1, \ldots, g_s \rangle$, and in fact it's not true in general.

Example. Let us consider $f = x^2y + xy^2 + y^2$, and $g_1 = xy - 1$, $g_2 = y^2 - 1$.

We have $\mathrm{LT}(f) = x^2y$. This is divisible by $\mathrm{LT}(g_1)$, so we can subtract xg_1 to get

$$(x^2y + xy^2 + y^2) - x(xy - 1) = xy^2 + x + y^2.$$

The new leading term is divisible by both $\mathrm{LT}(g_1)$ and $\mathrm{LT}(g_2)$; we use the first:

$$(xy^2 + x + y^2) - y(xy - 1) = x + y^2 + y.$$

Now the leading term isn't divisible by $\mathrm{LT}(g_1)$ or $\mathrm{LT}(g_2)$; so we leave it alone and go on to the next term y^2. We subtract $y^2 - 1$ to get

$$(x + y^2 + y) - (y^2 - 1) = x + y + 1.$$

There are no bad terms left, so $(x + y + 1)$ is the remainder, and we have computed

$$f = (x + y)(xy - 1) + (1)(y^2 - 1) + (x + y + 1).$$

is the remainder.

However, we made a choice at the second step: whether to use g_1 or g_1. What if we used g_2 instead? Then the computation goes

$$\begin{aligned}(x^2y + xy^2 + y^2) - x(xy - 1) &= xy^2 + x + y^2,\\ (xy^2 + x + y^2) - x(y^2 - 1) &= 2x + y^2,\\ (2x + y^2) - 1(y^2 - 1) &= 2x + 1.\end{aligned}$$

So we have $f = x(xy - 1) + (x + 1)(y^2 - 1) + (2x + 1)$, with a different remainder; both $x + y + 1$ and $2x + 1$ are valid possibilities for the remainder.

5.3bis: Monomial ideals

Mea culpa: I said some wrong things in the lecture at this point. The following is a careful attempt to explain away the resulting confusion.

In this course $0 \in \mathbb{N}$. Then, for any $n \geqslant 1$, the set $\mathbb{N}^n$ is a commutative *monoid* (a set with a commutative, associative binary operation – addition of vectors – and an identity element $(0,\ldots,0)$).

Definition. *A* monoid ideal *in* $\mathbb{N}^n$ *is a subset* $S \subseteq \mathbb{N}^n$ *with the property that* $m + s \in S$ *for all* $m \in \mathbb{N}^n$ *and* $s \in S$.

This is distinct from the following concept:

Definition. *A* monomial ideal *in* $R = F[X_1,\ldots,X_n]$ *is an ideal of R generated by monomials.*

These are different things – one is a subset of R, the other of $\mathbb{N}^n$ – but they are closely related. One checks that if I is a monomial ideal, and $f \in R$, then $f \in I$ iff every monomial in f is in I. From this, it is easy to show the following:

Proposition. *There is an inclusion-preserving bijection between monomial ideals* $I \trianglelefteqslant R$*, and monoid ideals* $S \subseteq \mathbb{N}^n$*, given as follows:*

- *if S is a monoid ideal, then the sub-F-vectorspace of R with basis* $\{X^s : s \in S\}$ *is a monomial ideal of R;*
- *if I is a monomial ideal, then* $S = \{s \in \mathbb{N}^n : X^s \in I\}$ *is a monoid ideal of* $\mathbb{N}^n$.

(We allow the zero ideal of R, which corresponds to the empty monoid ideal of $\mathbb{N}^d$*.)*

Moreover, given an arbitrary set of monomials T, the monomial ideal $\langle T\rangle$ of R corresponds to the monoid ideal $\bigcup_{t\in T}(t + \mathbb{N}^n)$, which is the monoid ideal of $\mathbb{N}^n$ generated by T. So it is straightforward to test if some $f \in R$ lies in $\langle T\rangle$: we just check if every monomial in f is divisible by some monomial in T.

5.4 Gröbner bases

Definition. *For* $I \trianglelefteqslant R$ *an ideal, we define* $\langle \mathrm{LT}(I)\rangle$ *to be the ideal of R generated by* $\{\mathrm{LT}(f) : f \in I - \{0\}\}$.

This is a monomial ideal, by definition; it corresponds to the monoid ideal $\{\operatorname{mdeg} f : f \in I - \{0\}\}$ of $\mathbb{N}^n$ (you should check that this is indeed a monoid ideal).

However, if $I = \langle f_1,\ldots,f_s\rangle$, it does *not* follow that $\mathrm{LT}(I) = \langle \mathrm{LT}(f_1),\ldots,\mathrm{LT}(f_s)\rangle$. Clearly $\mathrm{LT}(I)$ contains this, but it can be bigger:

Example. Let $I = \langle f_1, f_2 \rangle$ where $f_1 = x^3 - 2xy$ and $f_2 = x^2y - 2y^2 + y$. Then (for lexicographic order) we have $\mathrm{LT}(f_1) = x^3$, $\mathrm{LT}(f_2) = x^2y$. But we have

$$xf_2 - yf_1 = xy$$

and $\mathrm{LT}(xy) = xy \notin \langle x^3, x^2y \rangle$.

Definition. *Let $I \trianglelefteq R$. A* Gröbner basis *of I is a finite subset $G = \{g_1, \dots, g_s\} \subseteq I$ such that we have*

$$\langle \mathrm{LT}(g_1), \dots, \mathrm{LT}(g_s) \rangle = \langle \mathrm{LT}(I) \rangle.$$

Proposition. *Let I be an ideal. Then Gröbner bases of I exist. Moreover, any Gröbner basis of I is a generating set of I.*

Proof. The infinite set $\{\mathrm{LT}(f) : f \in I - \{0\}\}$ generates $\langle \mathrm{LT}(I) \rangle$, by definition. Since R is Noetherian, there must be a finite subset which generates the same ideal[2]. The corresponding elements of I form a Gröbner basis of I.

Now, given a Gröbner basis $G = g_1, \dots, g_s$, we can write any $f \in I$ as $\sum q_i g_i + r$, where no monomial in r is divisible by any of $\mathrm{LT}(g_1), \dots, \mathrm{LT}(g_s)$. Since $r \in I$ by construction, it follows that r cannot contain any monomials at all, i.e. we must have $r = 0$. Thus $f = \sum q_i g_i \in \langle G \rangle$. □

Corollary. *If G is a Gröbner basis of I, then for any $f \in R$, there is a* unique *r such that $f - r \in I$ and no monomial appearing in r is divisible by any of* $\mathrm{LT}(g_1), \dots, \mathrm{LT}(g_r)$*. Moreover, given G and f, we can algorithmically compute this remainder r, and some (non-unique) choice of q_i such that $f = \sum g_i q_i + r$.*

Proof. Suppose f had two different remainders r, r'. Then $r - r' \in I$; but no monomial appearing in either or r and r' can be in $\mathrm{LT}(I)$, so we must have $r - r' = 0$. The computability is clear from the above. □

So, **Gröbner bases solve the ideal membership problem**: given a Gröbner basis for I, we have $f \in I$ iff the remainder of f is zero, and we can test algorithmically if this is the case and if so, express f in terms of our generators. Of course, this is no use unless we can compute Gröbner bases to start with!

Remark. The notion of "leading term" and "Gröbner basis" still makes sense in $A[X_1, \dots, X_d]$ for any commutative ring A (not necessarily a field). The theory works well if A is a Euclidean domain, e.g. $\mathbb{Z}$; the theory of Gröbner bases for $\mathbb{Z}[X_1, \dots, X_n]$ would be a good project topic.

[2] We are using Hilbert's basis theorem here. This is morally the wrong way around: one of the simplest and cleanest proofs of the Hilbert basis theorem is actually to prove directly that any *monoid* ideal of $\mathbb{N}^d$ is finitely generated (Dickson's lemma), which suffices to prove that $\langle LT(I) \rangle$ is finitely generated, and then to use the second half of the proposition to deduce that I itself is finitely generated.

5.5 Buchberger's algorithm

Recall above our "bad" example of a generating set (g_1, g_2) that isn't a Gröbner basis; the problem came from $yg_1 - xg_2$ having smaller leading term than g_1 or g_2.

Definition. *Given two nonzero polynomials f, g, we define their S-**polynomial** by*

$$S(f,g) = \frac{X^\gamma}{\mathrm{LT}(f)} f - \frac{X^\gamma}{\mathrm{LT}(g)} g$$

where γ is the smallest exponent such that $\frac{x^\gamma}{\mathrm{LT}(f)}$ and $\frac{x^\gamma}{\mathrm{LT}(g)}$ are monomials.

This is designed to produce cancellation – its multidegree (the multidegree of its leading term with respect to $\preccurlyeq$) is smaller than expected, since the leading terms of the two summands cancel each other out. We're now going to show that "all cancellations among leading terms come from S-polynomials".

Lemma. *Let $p_1, \dots, p_s \in R$ all have the same multidegree δ. If $\sum_i p_i$ has multidegree $< \delta$, then $\sum_i p_i$ is a linear combination of the polynomials $S(p_i, p_j)$.*

Proof. Let $d_i \in F$ be the leading coefficient of p_i. Since all the p_i have the same degree, $S(p_i, p_j) = \frac{1}{d_i} p_i - \frac{1}{d_j} p_j$. If $\sum p_i$ has multidegree $< \delta$, we must have $\sum d_i = 0$; thus

$$\begin{aligned}\sum p_i &= d_1 \left(\tfrac{1}{d_1} p_1 - \tfrac{1}{d_s} p_s \right) + \cdots + d_{s-1} \left(\tfrac{1}{d_{s-1}} p_{s-1} - \tfrac{1}{d_s} p_s \right) + \left(p_s + \sum_{i=1}^{s-1} \tfrac{d_i}{d_s} p_s \right) \\ &= d_1 S(p_1, p_s) + \cdots + d_{s-1} S(p_{s-1}, p_s) + \tfrac{p_s}{d_s} \sum_{i=1}^{s} d_i,\end{aligned}$$

and the last term is 0 as we have seen. □

Theorem (Buchberger's criterion). *A finite set $G = \{g_1, \dots, g_s\}$ is a Gröbner basis of $I = \langle G \rangle$ if and only if, for all $i \neq j \in \{1, \dots, s\}$, the remainder of $S(g_i, g_j)$ on division by G is zero.*

Proof. One direction is obvious: $S(g_i, g_j) \in I$, so if G is a Gröbner basis, then the remainder must be 0.

Conversely, suppose G satisfies this criterion. Let $f \in I$, so $f = \sum g_i q_i$ for some q_i. We want to show that $\mathrm{LT}(f)$ is divisible by one of the $\mathrm{LT}(g_i)$. We may suppose that our choice of q_i is such that $\delta = \max_i(\mathrm{mdeg}(g_i q_i) : q_i \neq 0)$ is as small as possible (with respect to $\preccurlyeq$); this is possible, by the well-ordering property of $\preccurlyeq$.

If there is no cancellation in the sum, so $\mathrm{mdeg}\, f = \delta$, then $\mathrm{LT}(f)$ is divisible by $\mathrm{LT}(g_i)$ for some i, as required. We must show that if any cancellation occurs, then it contradicts the minimality assumption on the q_i.

Let $J = \{i \in 1, \dots, s : \operatorname{mdeg}(g_i q_i) = \delta\}$. Then

$$\begin{aligned} f &= \sum_{i \in J} g_i q_i + \sum_{i \neq J} g_i q_i \\ &= \sum_{i \in J} g_i \operatorname{LT}(q_i) + \sum_{i \in J} g_i (q_i - \operatorname{LT}(q_i)) + \sum_{i \notin J} g_i q_i \end{aligned}$$

The second and third terms only involve polynomials of multidegree $< \delta$, so if there is cancellation, it occurs in the first sum, which is a sum of polynomials of multidegree exactly δ. By the last Lemma, we can write

$$f = \sum_{i,j \in J} c_{ij} S\left(g_i \operatorname{LT}(q_i), g_j \operatorname{LT}(q_j)\right) + \text{smaller degree terms},$$

for some scalars $c_{ij} \in F$. However, since $\operatorname{LT}(q_i)$ and $\operatorname{LT}(q_j)$ are monomials, we have

$$S\left(g_i \operatorname{LT}(q_i), g_j \operatorname{LT}(q_j)\right) = x^{\delta - \gamma_{ij}} S(g_i, g_j),$$

where $\gamma_{ij} = \operatorname{lcm}(\operatorname{mdeg} g_i, \operatorname{mdeg} g_j)$.

By assumption, the division algorithm reduces each $S(g_i, g_j)$ to 0. So we can write

$$S(g_i, g_j) = \sum u_k g_k$$

for some u_k with $\operatorname{mdeg}(u_k g_k) \preccurlyeq \operatorname{mdeg} S(g_i, g_j) \prec \gamma_{ij}$ (strictly). Hence

$$S\left(g_i \operatorname{LT}(q_i), g_j \operatorname{LT}(q_j)\right) = \sum_k (x^{\delta - \gamma_{ij}} u_k) g_k, \quad \operatorname{mdeg}\left(x^{\delta - \gamma_{ij}} u_k\right) < \delta.$$

Substituting back, we obtain an expression for f in which all summands have multidegree $< \delta$, a contradiction. □

Buchberger's criterion gives an algorithmic test for whether G is a Gröbner basis. But it also gives more than that: if G is *not* a Gröbner basis, then one of the S-polynomials will not reduce to 0, meaning we have computed an explicit element of I whose leading term isn't in $\langle \operatorname{LT}(g_1), \dots, \operatorname{LT}(g_s) \rangle$. So if we enlarge G by adding the reduction of $S(g_i, g_j)$ to it, we get a new generating set G' for which the leading-term ideal $\langle \{\operatorname{LT}(g) : g \in G'\} \rangle$ is strictly larger. Since R is Noetherian, after finitely many iterations this process must stop, and we've computed a Gröbner basis for I. So we've shown the following:

Theorem (Buchberger's algorithm)**.** *Given a finite set of polynomials H generating an ideal I, we can compute a set $G \supseteq H$ which is a Gröbner basis for I, and express the elements of G as R-linear combinations of elements of H.*

5.6 Reduced Gröbner bases

Note that if G is a Gröbner basis of I, then so is any finite set $G' \supseteq G$ contained in I. Very often most of these generators are redundant:

Lemma. *Let G be a Gröbner basis of I. If $p \in G$ and $\mathrm{LT}(p)$ is divisible by $\mathrm{LT}(q)$ for some $q \in G - \{p\}$, then $G - \{p\}$ is a Gröbner basis of I.*

Proof. If $\mathrm{LT}(p)$ is divisible by some $\mathrm{LT}(q)$, then we have $\langle \mathrm{LT}(g) : g \in G \rangle = \langle LT(g) : g \in G - \{p\} \rangle$. By assumption the former is equal to $\langle \mathrm{LT}(I) \rangle$, hence so is the latter. □

We say a Gröbner basis is **minimal** if there is no $p \in G$ to which the above lemma applies. This is equivalent to requiring that no proper subset of G is a Gröbner basis of I. Given a Gröbner basis G of I, one can clearly find a minimal Gröbner basis of I, simply by throwing away any redundant basis elements until the result is minimal.

Exercise. Check that if G and G' are any two minimal Gröbner bases of the same ideal, then G and G' are the same size, and there is a bijection $\sigma : G \to G'$ such that $\sigma(g)$ and g have the same multidegree.

These aren't quite unique, e.g. $(X + Y, Y)$ and $(2X, 3Y)$ are Gröbner bases of the same ideal in $F[X, Y]$.

Definition. *A Gröbner basis G is* reduced *if, for all $p \in G$, the following holds:*

- *the leading coefficient of p is 1;*
- *no monomial of p is divisible by the leading term of any $q \in G - \{p\}$.*

If any $p \in G$ contains a (non-leading) monomial divisible by some other element of G, we can kill this monomial by adding an element of $\langle G - \{p\} \rangle$ to p (without changing $\mathrm{LT}(p)$, so the result is still a Gröbner basis of G); this process terminates after finitely many steps in a reduced Gröbner basis.

Proposition. *Every ideal of R has a* unique *reduced Gröbner basis.*

We won't prove this (although it isn't particularly hard, see Cox–Little–O'Shea theorem 2.7.5).

Remark. You should think of minimal Gröbner bases as being a bit like row echelon bases of vector spaces, and reduced Gröbner bases as being like RREF.

6 More computations with ideals

6.1 Elimination theory

Recall my "intersecting curves" puzzle from a couple of weeks back, which was to compute all the points in the finite set $V(\langle f_1, f_2\rangle)$ where

$$\begin{aligned} f_1 &= X(X+Y)(X+1) + Y(Y+1)(Y+X) + (X+1)(Y+1) - 4(X+Y)(Y+1)(X+1), \\ f_2 &= X^2 + Y^2 - 1. \end{aligned}$$

Let's hit it with the big Gröbner-basis hammer and see what we get.

```
sage: R.<X, Y> = PolynomialRing(QQ, order=’lex’)
sage: I = R.ideal( X*(X + Y)*(X + 1) + Y*(Y + 1)*(Y + X) + (X+1)*(Y+1) - 4*(X +
....: Y)*(Y + 1)*(X + 1), X^2 + Y^2 - 1 )
sage: I.groebner_basis()
[X + 112/23*Y^5 + 172/23*Y^4 - 273/46*Y^3 - 165/23*Y^2 + 109/46*Y + 1,
 Y^6 + 5/4*Y^5 - 23/32*Y^4 - 9/8*Y^3 - 1/32*Y^2 + 1/8*Y]
```

Let's look at that answer more carefully: the second polynomial **involves Y alone** – there are no X terms. So if (x, y) is an intersection point, then y is a root of this 1-variable sextic. Moreover, the other polynomial has the form $X + R(Y)$ for some one-variable polynomial; so the solutions are precisely $(-R(Y), Y)$ for Y a root of $Y^6 + \frac{5}{4}Y^5 + \dots$.

Definition. *For $1 \leqslant \ell < n$, let $R_\ell = R[X_{\ell+1}, \dots, X_n]$. For I an ideal in $R[X_1, \dots, X_n]$, the **i-th elimination ideal** of I is $I_\ell = I \cap R_\ell$, which is an ideal of R_ℓ. of R_ℓ. (Informally, it's the polynomials in I in which the first ℓ variables don't appear)*

Theorem (Elimination via Gröbner bases)**.** *Let G be a Gröbner basis of I with respect to lex ordering (with $x_1 \succ x_2 \succ \dots \succ x_n$). Then $G \cap R_\ell$ is a Gröbner basis of $I_\ell = I \cap R_\ell$ (and it is reduced if G is).*

Proof. Let $f \in I_\ell$. Then $\mathrm{LT}(f)$ has to be divisible by $\mathrm{LT}(g)$ for some $g \in G$. Thus $\mathrm{LT}(g) \in R_\ell$; but since any monomial not in R_ℓ is bigger than any monomial in R_ℓ, this forces all monomials in g to lie in R_ℓ. Hence $g \in G \cap R_\ell$. □

Remark. Note that this wouldn't work with an arbitrary monomial ordering; we need every monomial in R_ℓ to be smaller than every other monomial. That's why I had to explicitly write `order=’lex’` in the example: Sage's default monomial ordering is something else (the *degree reverse lexicographic* order), which doesn't have this elimination property.

6.2 Images of sets, preimages of ideals

Suppose we have a polynomial map $g : \overline{F}^m \to \overline{F}^n$, given in terms of coordinates $X_1, \dots, X_m$ and $Y_1, \dots, Y_n$ by $X = (X_1, \dots, X_m) \mapsto (g_1(X), \dots, g_n(X))$ for some $g_i \in F[X_1, \dots, X_m]$. I'm also going to suppose we have an F-algebraic set $S = V(J) \subseteq \overline{F}^m$; and I want to compute $g(S)$.

In general $g(S)$ isn't an algebraic set, but we can at least compute the smallest algebraic set containing it (the *Zariski closure* of $g(S)$), which is given by the ideal $I(g(S))$.

Proposition. *We have $I(g(S)) = (g^*)^{-1}(J)$, where*

$$g^* : F[Y_1, \dots, Y_n] \longrightarrow F[X_1, \dots, X_m]$$

is the F-algebra homomorphism sending $f(\underline{Y})$ to $f(g(\underline{X}))$, i.e. sending Y_j to $g_j(X_1, \dots, X_m)$ for each j.

Remark. The special case when J is the zero ideal, so we want to compute all polynomials in the Y's which vanish on the image of g, is called the *implicitization problem*: taking a subspace of $\overline{F}^n$ defined parametrically (as the image of g) and finding a description of it in terms of equations.

If our map g is the "forget the first ℓ variables" map from $\overline{F}^m \to \overline{F}^{m-\ell}$, then this is exactly the problem that elimination theory solves. What about the general case? We consider the diagram of morphisms

$$\overline{F}^m \longrightarrow \overline{F}^{(m+n)} \longrightarrow \overline{F}^n,$$

where the first map is $X \mapsto (X, g(X))$ (so the image is the graph of g), and the second is forgetting the first m variables. In terms of rings, the picture is

$$R[Y_1, \dots, Y_n] \longrightarrow R[X_1, \dots, X_m, Y_1, \dots, Y_n] \longrightarrow R[X_1, \dots, X_m]$$

where the first map is the obvious inclusion, and the second is given by sending the X's to themselves and sending each Y_j to $g_j(X_1, \dots, X_m)$.

Proposition. *If $I = \langle f_1, \dots, f_m \rangle$ is an ideal in $R[X_1, \dots, X_m]$, then the preimage of I in the ring $R[X_1, \dots, X_m, Y_1, \dots, Y_n]$ is the ideal*

$$\langle Y_1 - g_1, \dots, Y_n - g_n, f_1, \dots, f_m \rangle.$$

Proof. This is a straightforward check. □

Hence we can compute the preimage of I in $R[Y_1, \dots, Y_n]$ by finding a lex Gröbner basis for this ideal and hence eliminating the X's.

Example. Let's consider the map $F^2 \to F^3$ given by $(t, u) \mapsto (t(u^2 - t^2), u, u^2 - t^2)$. What's a minimal set of equations describing its image?

Using the above arguments, we want to eliminate (t, u) from the ideal of $R[t, u, x, y, z]$ given by

$$I = \langle x - t(u^2 - t^2), y - u, z - u^2 - t^2 \rangle.$$

The Gröbner basis (for lex ordering with $t \succ u \succ x \succ y \succ z$) is

```
sage: R.<t, u, x, y, z> = PolynomialRing(QQ, order='lex')
sage: I = R.ideal([x - t*(u^2 - t^2), y - u, z - (u^2 - t^2)])
sage: I.groebner_basis()
[t^2 - y^2 + z, t*x - y^2*z + z^2, t*z - x, u - y, x^2 - y^2*z^2 + z^3]
```

So the ideal of polynomials vanishing on the image of $\overline{F}^2$ is generated by $x^2 - y^2z^2 + z^3$. We can also get this in one step using

```
sage: I.elimination_ideal([t, u])
Ideal (x^2 - y^2*z^2 + z^3) of Multivariate Polynomial Ring in t, u, x, y, z
 over Rational Field
```

(The answer is nonsense, formally, since we're looking for an ideal of $F[x, y, z]$ not $F[t, u, x, y, z]$ – a long-standing bug in Sage. But the list of generators is correct.)

6.3 Dimensions and Hilbert polynomials

Here's a basic fact from algebraic geometry:

Proposition. *Let $R^{\leqslant d}$ denote the F-subspace of $R = F[X_1, \dots, X_n]$ spanned by monomials of total degree $\leqslant d$. Then for any ideal $I \trianglelefteq R$, the function*

$$HF_I(d) = \dim_F \left(\frac{R_{\leqslant d}}{R_{\leqslant d} \cap I} \right)$$

is eventually polynomial*: there is a polynomial $HP_I(d) \in \mathbb{Q}[d]$ and a $d_0 \in \mathbb{N}$ such that $HF_I(d) = HP_I(d)$ for all $d \geqslant d_0$.*

We call HP_I the *affine Hilbert polynomial* of I. It encodes various geometric properties of the algebraic set $V = V(I)$; for instance, if HP_I has degree r, then r is the dimension of V.

Example. HP_I is constant if and only if R/I is finite-dimensional over F, which is equivalent to V being a finite set (and the constant is equal to the number of points in $V(I)$, counted with an appropriate multiplicity).

Remark. In the literature one more often encounters *projective Hilbert polynomials*, which correspond to algebraic subsets of projective space (rather than affine space). We'll discuss projective Hilbert polynomials in a moment.

It turns out we can compute Hilbert polynomials using Gröbner bases, but only for a special kind of monomial order:

Definition. *A monomial order is* graded *if it extends the partial ordering given by total degree, so we always have $M \prec N$ if M has lower total degree than N.*

Lex-order is not a graded order (unless $n = 1$). The standard example is "degree lexicographic" order, which you saw on the exercise sheets:

$$M \prec_{\text{deglex}} N \Leftrightarrow (\text{tot. deg. } M < \text{tot. deg. } N) \text{ or } (\text{tot. deg. } M = \text{tot. deg. } N \text{ and } M \prec_{\text{lex}} N).$$

Proposition. *Let I be an ideal and let $\langle LT(I)\rangle$ be its leading term ideal for a graded monomial order. Then we have*

$$HF_I = HF_{\langle LT(I)\rangle}.$$

We won't prove this here; see C-L-O'S §9.3 for the proof.

Computing Hilbert polynomials for monomial ideals is pretty explicit: we're just counting the monomials that *aren't* divisible by any of the generators, and we can do this using the inclusion-exclusion principle, using the fact that there are $\binom{d+n}{n}$ monomials of degree $\leqslant d$ in n variables:

Example. For the ideal $\langle X^2Y^3, XZ\rangle$ of $F[X, Y, Z]$, we have

$$\begin{aligned} HP_I = \#\{&\text{all monomials of deg} \leqslant d\} \\ &- \#\{\text{monomials of deg} \leqslant d \text{ divisible by } X^2Y^3\} \\ &- \#\{\text{monomials of deg} \leqslant d \text{ divisible by } XZ\} \\ &+ \#\{\text{monomials of deg} \leqslant d \text{ divisible by both } X^2Y^3 \text{ and } XZ\} \end{aligned}$$

Now, a monomial is divisible by X^2Y^3 and XZ iff it's divisible by X^2Y^3Z, so we can write this as

$$\begin{aligned} HP_I = \#\{&\text{all monomials of deg} \leqslant d\} \\ &- \#\{\text{monomials of deg} \leqslant d \text{ divisible by } X^2Y^3\} \\ &- \#\{\text{monomials of deg} \leqslant d \text{ divisible by } XZ\} \\ &+ \#\{\text{monomials of deg} \leqslant d \text{ divisible by } X^2Y^3Z\}. \end{aligned}$$

If d is large enough, the monomials of degree $\leqslant d$ divisible by X^2Y^3 are precisely the products X^2Y^3M where M has degree $\leqslant d - 5$, so this is

$$\begin{aligned} HP_I = \#\{&\text{all monomials of deg} \leqslant d\} \\ &- \#\{\text{monomials of deg} \leqslant d-5\} \\ &- \#\{\text{monomials of deg} \leqslant d-2\} \\ &+ \#\{\text{monomials of deg} \leqslant (d-6)\}. \end{aligned}$$

$$= \binom{d+3}{3} - \binom{d+3-5}{3} - \binom{d+3-2}{3} + \binom{d+3-6}{3} = \frac{(d^2 + 11d - 10)}{2}.$$

The general formula is

Proposition. *If $M_1, \dots, M_s$ are monomials, then*

$$HP_{\langle M_1,\dots,M_s\rangle}(d) = \sum_{T\subseteq\{1,\dots,s\}} (-1)^{\#T} \binom{d + n - \deg M_T}{n},$$

where $M_T = \text{LCM}\{M_j : j \in T\}$.

Combining this with the previous proposition gives an algorithm to compute the dimension of any algebraic set. In Sage you can get this with the `dimension` method:

```
sage: R.<X, Y, Z> = QQ[]
sage: I = R.ideal([X^3 - Y*Z, X^2*Y - Z^2, Y^2 - X*Z])
sage: I.dimension()
1
```

(This is an example of a 1-dimensional variety in 3-dimensional space that can't be cut out by just 2 polynomials.)

$6.4\frac{1}{2}$. Digression: Affine and projective varieties

Definition. *For $n \geqslant 1$ and any field K, we define* projective n-space *over K as*

$$\mathbb{P}^n(K) = \{(x_0, \ldots, x_n) \in K^{n+1} : \exists i \text{ such that } x_i \neq 0\}/K^\times.$$

We embed K^n into $\mathbb{P}^n(K)$ as $(x_1, \ldots, x_n) \mapsto (1, x_1, \ldots, x_n)$.

There is a notion of *F-algebraic sets* in $\mathbb{P}^n(\overline{F})$, which correspond to *homogenous* ideals of $F[X_0, \ldots, X_n]$. An ideal is *homogenous* if it is generated by homogenous polynomials (polynomials in which all monomials have the same total degree). Note that we don't require all the generators of an ideal to be homogenous of the same degree, so

$$\langle x_0 + x_1, x_0 x_1 \rangle \text{ is homogenous, but } \langle x_0, x_1 + 2 \rangle \text{ is not.}$$

For a homogenous ideal one can define its *projective Hilbert polynomial* by

$${}^p HF_I(d) = \dim \frac{R_{=d}}{I \cap R_{=d}}.$$

Note we have $=$, where before we had $\leqslant$.

Sage (and its competitors e.g. Magma), have one-line commands for projective Hilbert polynomials, but annoyingly not for affine ones. This can be got around using *homogenization*.

Definition. *Let $f \in F[X_1, \ldots, X_n]$ of total degree d. Then its* ***homogenization*** *f^h is*

$$f^h = X_0^d f(\tfrac{X_1}{X_0}, \ldots, \tfrac{X_n}{X_0}) \in F[X_0, X_1, \ldots, X_n].$$

The homogenization of an ideal I is $I^h = \langle \{f^h : f \in I\} \rangle$.

Geometrically, this corresponds to passing from an algebraic set $V \subset \overline{F}^n$ to its closure $\overline{V} \subset \mathbb{P}^n(\overline{F})$.

Proposition. *We have ${}^p HF_{I^h} = HF_I$, and the same for HP.*

Proof. One checks that setting $X_0 = 1$ gives an isomorphism

$$\tilde{R}_{=d}/(I^h \cap \tilde{R}_{=d}) \cong R_{\leqslant d}/(I \cap R_{\leqslant d}).$$

So both spaces have the same dimension. □

A slight subtlety is that if G is a generating set of I, it's not obvious that $\{g^h : g \in G\}$ generates I^h, and in fact it's false in general. However, it's true for Gröbner bases:

Proposition. *Let G be a GB of I with respect to any graded monomial order. Then $\{g^h : g \in G\}$ generates I^h.*

Putting this together, we can compute affine Hilbert polynomials as follows:

```
sage: R.<X, Y, Z> = QQ[] # use default degrevlex order (which is graded)
sage: I = R.ideal([X^2 * Y^3, X * Z])
sage: Rh.<T, X, Y, Z> = QQ[]
sage: Ih = Rh.ideal([f.homogenize() for f in I.groebner_basis()])
sage: L.hilbert_polynomial()
1/2*t^2 + 11/2*t - 5
```

6.4 Solving equations

If HP_I is a constant, then $V(I)$ is a finite set. Hence, for each $1 \leqslant j \leqslant n$, the intersection $I \cap F[X_j]$ is generated by some nonzero polynomial h_j.

Proposition. *Let $\hat{h}_j$ be the square-free part of h_j. Then* $\operatorname{rad} I = \langle I, \hat{h}_1, \dots, \hat{h}_n \rangle$.

Proof. We have

$$R/\langle \hat{h}_1, \dots, \hat{h}_n \rangle = \bigotimes_{i=1}^{n} \frac{F[X_j]}{\hat{h}_j(X_j)}$$

is isomorphic to a finite direct product of field extensions of F. Hence any quotient of this ring is also a finite direct product of fields, and so cannot have any nilpotent elements. □

Note that if I is radical and zero-dimensional then HP_I is a constant, and this constant is exactly $\#V(I)$. It remains to actually compute $V(I)$ itself. This can be done rather efficiently using lex Gröbner bases: once we have the Gröbner basis, we can read off all the elimination ideals I_ℓ, and hence find all possibilities for X_n, all possibilities for X_{n-1} for a given X_n, etc – just like solving linear equations using a triangular form of the matrix.

Example. Consider the equations

$$\begin{aligned} x^2 + y + z &= 1, \\ x + y^2 + z &= 1, \\ x + y + z^2 &= 1. \end{aligned}$$

We compute the lex Gröbner basis of $J = \langle x^2 + y + z - 1, x + \dots \rangle$ in Sage and get

$$\begin{array}{r} x + y + z^2 - 1 = 0, \\ \hline y^2 - y - z^2 + z = 0, \\ yz^2 + \frac{1}{2}z^4 - \frac{1}{2}z^2 = 0, \\ \hline z^6 - 4z^4 + 4z^3 - z^2 = 0. \end{array}$$

Remark. Note that this is actually larger than the original generating set, so Gröbner bases aren't literally the "simplest possible generating sets" in a naive sense.

The last polynomial has roots $0, 1, -1 \pm \sqrt{2}$, and we can work back for each of them to get the solutions in $\mathbb{Q}$, which are

$$(0,0,1), (0,1,0), (1,0,0), \text{ and } (\alpha, \alpha, \alpha) \text{ where } \alpha = -1 \pm \sqrt{2}.$$

Note this consists of exactly 5 points, and the affine Hilbert polynomial of $\operatorname{rad} J$ is just the constant 5, which fits. (The Hilbert polynomial of J itself is the constant 8, because the first three points have "multiplicity 2" – the local ring of R/I at these points is 2-dimensional.)

7 Algebraic number theory

7.1 Number fields

A *number field*, for us, means a field of the form $K = \mathbb{Q}(\alpha)$, where α is a root of an irreducible polynomial $f \in \mathbb{Q}[X]$.

We can think of α as living in $\mathbb{C}$ if we like; but for computations it's generally better to think of α as a formal symbol, i.e. we identify our field with the quotient ring $\mathbb{Q}[X]/\langle f(X)\rangle$ and α is just the image of X modulo f. In this setup, each root of f in $\mathbb{C}$ defines an *embedding* $K \hookrightarrow \mathbb{C}$, but we don't have to fix any one of these embeddings as being "better" than the rest.

Of course, we can have many different polynomials generating the same field; so we'd like algorithms to check for this. That comes as a corollary of the following:

Lemma. *Given a nonzero polynomial $g \in K[X]$, we can compute a factorization of g into irreducibles in $K[X]$.*

Proof (Trager's algorithm). Consider the quotient ring $R = K[X]/g(X)$. This is a $\mathbb{Q}$-algebra of dimension $\deg(f) \times \deg(g)$, which we can identify with $\mathbb{Q}[X_1, X_2]/(f(X_1), g(X_2))$; it is isomorphic to a finite product of field extensions of K, one for each irreducible factor of g.

If we pick a random $y \in R$, then we can compute the powers $1, y, \dots, y^{\deg(f)\times\deg(g)-1}$ and check whether they're linearly independent over $\mathbb{Q}$. The y for which this fails are contained in a finite union of proper $\mathbb{Q}$-linear subspaces of R, so we will eventually find a y which works. Hence we can compute the minimal polynomial of y and thus find an isomorphism $R \cong \mathbb{Q}[y]/q(y)$; and since X and α are in R, we can write them as polynomials in y, using linear algebra.

If we factor q into irreducibles in $\mathbb{Q}[y]$, then we get a decomposition of R as a product of field extensions of $\mathbb{Q}$, say $q_1(y), \dots, q_r(y)$. Thus $R \cong \prod_j \mathbb{Q}[y]/q_j(y)$, which must coincide with the product decomposition above. Moreover, we can find the images of α and x in each factor, and hence identify it as $K[X]/g_j(X)$ for some g_j. □

Remark. There's nothing special about $\mathbb{Q}$ here – this shows that if we can factor polynomials into irreducibles over an infinite field F, we can do it over any finite extension E/F. (Finite fields are a little troublesome, since we may not be able to find any sufficiently generic y, but there are other, better algorithms for the finite-field case.)

7.2 Rings of integers

The real depth of algebraic number theory comes when we think about how the integers embed in K.

Definition. *An element $\beta \in K$ is an* algebraic integer *if there exists a monic polynomial $g \in \mathbb{Z}[X]$ such that $g(\beta) = 0$.*

This is equivalent to asking that the minimal polynomial of β is integral, so we can easily test if a given x is integral.

Definition. *The* ring of integers $\mathcal{O}_K$ *is $\{x \in K : x \text{ is an algebraic integer }\}$.*

This is (nonobviously) a subring, as the name suggests. It is isomorphic to $\mathbb{Z}^d$ as an abelian group, and spans K over $\mathbb{Q}$, so any $x \in K$ can be written (computably) as y/n for $y \in \mathcal{O}_K$ and $n \in \mathbb{N}_{\geqslant 1}$.

Example. The golden ratio $\phi = \frac{1+\sqrt{5}}{2}$ is an algebraic integer (despite the 2 in the denominator), since it satisfies $x^2 = x + 1$.

Problem. *Given an irreducible polynomial f defining a number field K, can we compute a $\mathbb{Z}$-basis for $\mathcal{O}_K$?*

One simple approach to this is as follows. Scaling the variable appropriately, we can assume f is monic and integral. So the ring $A = \mathbb{Z}[\alpha] = \mathbb{Z} + \mathbb{Z}\alpha + \cdots + \mathbb{Z}\alpha^{d-1}$ is contained in $\mathcal{O}_K$, and it must have finite index as a subgroup of $\mathcal{O}_K$, since both rings span K over $\mathbb{Q}$. Finite-index subrings of $\mathcal{O}_K$ are called *orders* in K, so $\mathbb{Z}[\alpha]$ is an example of an order.

Starting from any order A, we'll now show how to "grow" A until we fill up the full ring of integers.

Definition.

(i) The trace map $\mathrm{tr} : K \to \mathbb{Q}$ *is the $\mathbb{Q}$-linear map sending x to the trace of the $d \times d$ matrix over $\mathbb{Q}$ giving the multiplication-by-x on K. This restricts to a map $\mathcal{O}_K \to \mathbb{Z}$.*

(ii) If A is an order in K, with basis $m_1, \ldots, m_d$, then the discriminant $\Delta(A)$ *is the determinant of the $d \times d$ matrix* $\mathrm{tr}(m_i m_j)$*, which is independent of the choice of basis [why? – exercise].*

Proposition. *If A is any order in K, then $\Delta(A) \in \mathbb{Z}$, and we have*

$$\Delta(A) = [\mathcal{O}_K : A]^2 \Delta(\mathcal{O}_K).$$

Since Δ_K is also an integer, if we can compute $\Delta(A)$ then we can give an upper bound for the index $[\mathcal{O}_K : A]$. In particular, if $\Delta(A)$ is square-free we must have $A = \mathcal{O}_K$ (although the converse is false.) In any case, this means $\mathcal{O}_K$ must be one of a finite list of possibilities; and we can (in principle) enumerate them all, and check which is the largest one whose generators are all algebraic integers. This must be $\mathcal{O}_K$.

Remark. For the linear algebra computations, we identify K with $\mathbb{Q}^d$ via a choice of defining polynomial, giving a basis $1, \alpha, \dots, \alpha^{d-1}$ of K as a $\mathbb{Q}$-vector space. We can then store $\mathcal{O}_K$ by computing Hermite-form basis vectors for $\mathcal{O}_K$ as a $\mathbb{Z}$-module. Note that the image of $\mathcal{O}_K$ in $\mathbb{Q}^d$ will generally be *larger* than $\mathbb{Z}^d$ itself, so we need to be able to work with $\mathbb{Z}$-submodules of $\mathbb{Q}^d$ which aren't contained in $\mathbb{Z}^d$.

Definition. *We define the* discriminant of *K to be $\Delta_K = \Delta(\mathcal{O}_K)$.*

Example. Let D be a square-free integer (other than 1). Let's compute the discriminant of $\mathbb{Q}(\sqrt{D})$.

An obvious choice of α is $\sqrt{D}$ itself. Then the trace-pairing matrix for $\mathbb{Z}[\sqrt{D}]$, in the basis $(1, \sqrt{D})$, is $\begin{pmatrix} 2 & 0 \\ 0 & 2D \end{pmatrix}$; so $\Delta(\mathbb{Z}[\sqrt{D}]) = 4D$. Since D is squarefree, we immediately see that $[\mathcal{O}_K : \mathbb{Z}[\sqrt{D}]]$ must be 1 or 2.

If the index is 2, then $\mathcal{O}_K$ has to contain one of $\frac{1}{2}, \frac{\sqrt{D}}{2}$, and $\frac{1+\sqrt{D}}{2}$. The first two are never algebraic integers, while the second has minimal polynomial $X^2 - X + \frac{1-D}{4}$, so it's an algebraic integer iff $D = 1 \bmod 4$. So

$$\Delta_{\mathbb{Q}(\sqrt{D})} = \begin{cases} D & \text{if } D = 1 \bmod 4, \\ 4D & \text{otherwise,} \end{cases}$$

and the Hermite-form basis matrix is either

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Remark. Note that for quadratic fields we always have $\mathcal{O}_K = \mathbb{Z}[\alpha]$ for *some* $\alpha \in K$. A number field with this property is calld *monogenic*. Most number fields of degree > 2 are not monogenic (e.g. $\mathbb{Q}(\sqrt{7}, \sqrt{10})$ is not monogenic).

Implementations

We can compute integer rings of number fields in Sage:

```
sage: K.<a> = NumberField(x^2 - 5)
sage: K.ring_of_integers()
Maximal Order in Number Field in a with defining polynomial x^2 - 5
sage: OK = _
sage: OK.free_module()
Free module of degree 2 and rank 2 over Integer Ring
User basis matrix:
[1/2 1/2]
[  0   1]
sage: L.<b> = NumberField(x^8 - 4*x^6 + 14*x^4 - 8*x^2 + 4)
sage: L.ring_of_integers()
```

```
Maximal Order in Number Field in b with defining polynomial x^8 - 4*x^6 + 14*x^4 - 8*x^2 + 4
sage: L.ring_of_integers().free_module()
Free module of degree 8 and rank 8 over Integer Ring
User basis matrix:
[ 1/7    0    0    0    0    0 5/14    0]
[   0  1/7    0    0    0    0    0 5/14]
[   0    0    1    0    0    0    0    0]
[   0    0    0    1    0    0    0    0]
[   0    0    0    0  1/2    0    0    0]
[   0    0    0    0    0  1/2    0    0]
[   0    0    0    0    0    0  1/2    0]
[   0    0    0    0    0    0    0  1/2]
```

In fact Sage is outsourcing the actual computation to another program called PARI/GP, a special-purpose package for number theory. You can get at this directly by typing `sage -gp` instead of `sage`:

```
$ sage -gp
                                               GP/PARI CALCULATOR Version 2.13.3 (released)
                                       amd64 running linux (x86-64/GMP-6.2.1 kernel) 64-bit
                            compiled: Jan 31 2022, gcc version 5.4.0 20160609 (Ubuntu 5.4.0-
                                                         threading engine: pthread
                                              (readline v6.3 enabled, extended help enabled)

                                                  Copyright (C) 2000-2020 The PARI Group

PARI/GP is free software, covered by the GNU General Public License, and comes WITHOUT ANY W

Type ? for help, \q to quit.
Type ?17 for how to get moral (and possibly technical) support.

parisize = 8000000, primelimit = 500000, nbthreads = 72
? nfbasis(x^2 - 5)
%1 = [1, 1/2*x - 1/2]
? nfbasis(x^8 - 4*x^6 + 14*x^4 - 8*x^2 + 4)
%17 = [1, x, x^2, x^3, 1/2*x^4, 1/2*x^5, 1/14*x^6 + 3/7, 1/14*x^7 + 3/7*x]
```

I find that for number field computations, Sage is a lot more user-friendly, but its scope is limited; there are lots of things that PARI can do which Sage doesn't offer.

7.3 Ideals and factorization

7.3.1 History

The original motivation for studying algebraic number fields was in order to solve equations in integers – and, in particular, Fermat's last theorem. If you introduce an n-th root of unity ζ, then Fermat's equation becomes

$$Y^n = (X - Z)(X - \zeta Z) \dots (X - \zeta^{n-1} Z),$$

and if you have a good theory of *prime factorisation* in these rings, then you can get somewhere.

Sadly it's not quite so easy, because the rings $\mathcal{O}_K$ are typically **not unique factorization domains**. The great insight of Kummer was that there is a good theory for *ideals* of integer rings, and the failure of unique factorization of *elements* is because of the presence of non-principal prime ideals. So studying ideals in number fields is important.

Theorem. *Any (nonzero) ideal $\mathfrak{a} \trianglelefteq \mathcal{O}_K$ can be written as a product $\mathfrak{p}_1^{n_1}\mathfrak{p}_2^{n_2} \dots \mathfrak{p}_k^{n_k}$ for some $k \geqslant 0$, where $\mathfrak{p}_i$ are distinct prime ideals and $n_i \geqslant 1$; and this expression is unique up to the ordering of the factors.*

For a proof, see any book on algebraic number theory (e.g. Stewart + Tall, Neukirch, or Fröhlich + Taylor). We really need $\mathcal{O}_K$ to be the full ring of integers here – the statement would be false for a non-maximal order.

Proposition. *If K is a number field, then $\mathcal{O}_K$ is a unique factorization domain iff it is a principal ideal domain.* □

Example. We'll see shortly that in $\mathbb{Z}[\sqrt{-6}]$, the ideals $\langle 5 \rangle$ and $\langle 11 \rangle$ each factor as a product of two distinct prime ideals,

$$\langle 5 \rangle = \mathfrak{p}_1\mathfrak{p}_2, \qquad \langle 11 \rangle = \mathfrak{q}_1\mathfrak{q}_2.$$

The ideals $\mathfrak{p}_i$ and $\mathfrak{q}_i$ are all non-principal, but we can group them in two ways to get principal ideals,

$$\langle 55 \rangle = (\mathfrak{p}_1\mathfrak{p}_2)(\mathfrak{q}_1\mathfrak{q}_2) = (\mathfrak{p}_1\mathfrak{q}_1)(\mathfrak{p}_2\mathfrak{q}_2)$$

which correspond to two distinct factorizations of 55 into irreducible elements of $\mathcal{O}_K$,

$$55 = 5 \times 11 = (1 - 3\sqrt{-6}) \times (1 + 3\sqrt{-6}),$$

showing that $\mathbb{Z}[\sqrt{-6}]$ is not a UFD.

7.3.2 Computing with ideals

For computations, we fix a choice of defining polynomial f and hence identify K with $\mathbb{Q}^d$ as a vector space, using $(1, \alpha, \dots, \alpha^{d-1})$ as a basis. I'll assume we've already computed the Hermite-form $\mathbb{Z}$-basis for $\mathcal{O}_K$.

An ideal is just a special kind of $\mathbb{Z}$-submodule of K, so we can also represent it by a HNF basis. If we are given $x_1, \dots, x_r \in \mathcal{O}_K$, we can compute $\mathbb{Z}$-module generators for the ideal $\langle x_1, \dots, x_r \rangle$ by considering all possible pairs $x_i v_j$ where $v_1, \dots, v_n$ are $\mathbb{Z}$-module generators of $\mathcal{O}_K$. As a special case, this tells us if a $\mathbb{Z}$-submodule given in HNF is an ideal or not (we test if it coincides with the ideal generated by its $\mathbb{Z}$-basis).

Similarly, we can compute products of ideals: if $a_1, \dots, a_n$ are a $\mathbb{Z}$-basis of $\mathfrak{a}$, and $b_1, \dots, b_n$ of $\mathfrak{b}$, then the products $a_i b_j$ span $\mathfrak{ab}$, and we can extract a HNF basis by linear algebra.

Dividing by ideals is more subtle. From the Factorization Theorem we see that if $\mathfrak{a} \supseteq \mathfrak{b}$ then there is an ideal $\mathfrak{c}$ such that $\mathfrak{ac} = \mathfrak{b}$. Clearly we must have

$$\mathfrak{c} = \{c \in \mathcal{O}_K : ac \in \mathfrak{b} \text{ for all } a \in \mathfrak{a}\} = \bigcap_i a_i^{-1}\mathfrak{b},$$

where a_i are any generators of $\mathfrak{a}$ as an ideal. This is clearly computable.

(Caution: it is obvious that $\mathfrak{c}$ is an ideal and $\mathfrak{ac} \subseteq \mathfrak{b}$; it is much less obvious that $\mathfrak{ac} = \mathfrak{b}$, and this would fail if we worked with ideals of non-maximal orders. Proving that this ideal quotient is well-behaved is one of the key steps in proving the factorization theorem.)

7.3.3 Norms of ideals

Definition. *The* norm *of a nonzero ideal* $\mathfrak{a}$*, written* $N(\mathfrak{a})$*, is the order of the quotient ring* $\mathcal{O}_K/\mathfrak{a}$*.*

Here are some properties of the norm map:

- It's multiplicative, $N(\mathfrak{ab}) = N(\mathfrak{a})N(\mathfrak{b})$.
- If $N(\mathfrak{a}) = n$, then multiplication by n kills the quotient $\mathcal{O}_K/\mathfrak{a}$, so $n \in \mathfrak{a}$.
- If $N(\mathfrak{a}) = n$ then $\mathfrak{a}$ divides $\langle n \rangle$, and the factorization theorem shows that there are only finitely many possibilities for $\mathfrak{a}$; so there are **only finitely many ideals of a given norm**.
- If $\mathfrak{a} = \langle x \rangle$ is principal, then $N(\mathfrak{a}) = |N(x)|$, where $N(x)$ is the usual field norm $K^\times \to \mathbb{Q}^\times$.
- We can compute $N(\mathfrak{a})$ easily from a Hermite-form basis matrix of $\mathfrak{a}$.

Example. In $\mathbb{Q}(\sqrt{-6})$, the ring of integers is $\mathbb{Z}[\sqrt{-6}]$. Consider the ideal $\mathfrak{p}_1 = \langle 5, 2 + \sqrt{-6} \rangle$. This is spanned as a $\mathbb{Z}$-module by

$$\{5,\ 5\sqrt{-6},\ 2 + \sqrt{-6},\ 2\sqrt{-6} - 6\}$$

which we can express as vectors in the integral basis $\{1, \sqrt{-6}\}$ as

$$\begin{pmatrix} 5 & 0 \\ 0 & 5 \\ 2 & 1 \\ -6 & 2 \end{pmatrix}, \qquad \text{RREF} = \begin{pmatrix} 1 & 3 \\ 0 & 5 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

So this is an index 5 submodule of $\mathcal{O}_K$, i.e. the norm is 5. Any proper factor of $\mathfrak{p}$ would have to have norm a proper factor of 5, contradiction; so in fact $\mathfrak{p}$ is prime. Similarly, $\mathfrak{p}_2 = \langle 5, 2 - \sqrt{-6} \rangle$ is also a prime of norm 5, and its HNF basis is $\{(1, 2), (0, 5)\}$ so $\mathfrak{p}_1 \neq \mathfrak{p}_2$.

I claim $\mathfrak{p}_1$ is **not principal**. Suppose it were; then its generator would have to have norm ± 5. But the norm of $a + b\sqrt{-6}$ is $a^2 + 6b^2$, and $a^2 + 6b^2 = \pm 5$ clearly has no solutions in integers. So $\mathfrak{p}_1$ (and likewise $\mathfrak{p}_2$) are non-principal ideals.

7.3.4 Prime ideals and Dedekind–Kummer

Let's focus on prime ideals for a moment. Here are some general algebraic facts about prime ideals:

Proposition. *If $\mathfrak{p}$ is prime, then $\mathfrak{p} \cap \mathbb{Z} = p\mathbb{Z}$ for some prime $p \in \mathbb{N}$; we say $\mathfrak{p}$ **lies above** p.*

If $\mathfrak{p}$ lies above p, then $N(\mathfrak{p})$ is a power of p (it is p^f where $f \in \mathbb{Z}_{\geqslant 1}$ is the field extension degree $[\mathcal{O}_K/\mathfrak{p} : \mathbb{F}_p]$).

Given p, there are only finitely many prime ideals lying above p, and if $\mathfrak{p}_1, \dots, \mathfrak{p}_g$ are these primes, with norms $p^{f_1}, \dots, p^{f_g}$, then we have

$$\langle p \rangle = \prod_i \mathfrak{p}_i^{e_i}, \qquad e_i \text{ integers } \geqslant 1, \qquad \sum e_i f_i = [K : \mathbb{Q}].$$

If p does not divide Δ_K, then all the e_i are 1. □

Now, how do we compute the primes of $\mathcal{O}_K$ above a given prime p?

Proposition (Dedekind–Kummer theorem). *Let p be prime, and let $\alpha \in \mathcal{O}_K$ be such that p doesn't divide $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$. Let f be the minimal polynomial of α.*

If $\prod_i \bar{f}_i^{e_i}$ is the prime factorization of $\bar{f} = f \bmod p$ in $\mathbb{F}_p[X]$, then the prime factorisation of the ideal $\langle p \rangle$ in $\mathcal{O}_K$ is given by $\prod_i \langle p, f_i(\alpha) \rangle^{e_i}$, where $f_i \in \mathbb{Z}[X]$ is any lifting of $\bar{f}_i$.

Proof sketch. For simplicity suppose $\mathcal{O}_K = \mathbb{Z}[\alpha]$. Then the result follows by thinking about the quotient $\mathbb{Z}[X]/\langle p, f(X) \rangle$ in two ways: either as $(\mathbb{Z}[X]/f)/p = \mathcal{O}_K/p$, or as $(\mathbb{Z}[X]/p)/f = \mathbb{F}_p[X]/\bar{f}(X)$. So we get a one-to-one, inclusion-preserving correspondence between ideals of $\mathcal{O}_K$ containing p, and ideals of $\mathbb{F}_p[X]$ containing $\bar{f}$. □

Example. If $K = \mathbb{Q}(\sqrt{-6})$ and $\alpha = \sqrt{-6}$, then $\mathcal{O}_K = \mathbb{Z}[\alpha]$, so any prime p is allowed.

Mod 5 we have $X^2 + 6 = X^2 - 4 = (X-2)(X+2)$ and we get the factorization

$$\langle 5 \rangle = \langle 5, 2 + \sqrt{-6} \rangle \langle 5, 2 - \sqrt{-6} \rangle$$

from before.

For $p = 13$, $X^2 + 6$ is irreducible in $\mathbb{F}_{13}[X]$, so $\langle 13 \rangle$ is prime in $\mathbb{Z}[\sqrt{-6}]$.

For $p = 2$, $X^2 - 6 = X^2$ so $\langle 2 \rangle = \langle 2, \sqrt{-6} \rangle^2$.

7.3.5 Two-element generating sets

A by-product of the above proof is that all prime ideals (except possibly those dividing $[\mathcal{O}_K : \mathbb{Z}[\alpha]]$) can be generated by at most 2 elements, one of which is in $\mathbb{Z}$. This is no accident:

Proposition. *Let $\mathfrak{a}, \mathfrak{b} \trianglelefteq \mathcal{O}_K$ be nonzero ideals with $\mathfrak{b} \subseteq \mathfrak{a}$. Then there exists $y \in \mathfrak{a}$ such that $\langle \mathfrak{b}, y \rangle = \mathfrak{a}$.*

Proof. This is equivalent to showing that all ideals of the quotient ring $R = \mathcal{O}_K / \mathfrak{b}$ are principal. Note that this ring is finite.

If $\mathfrak{b} = \prod \mathfrak{p}_i^{m_i}$, then $R \cong \prod_i (\mathcal{O}_K / \mathfrak{p}_i^{m_i})$, so we can assume $\mathfrak{b} = \mathfrak{p}^m$ is a prime power. In this case, the ideals of R are just $\mathfrak{P}^i$ for $1 \leqslant i \leqslant m$, where $\mathfrak{P}$ is the image of $\mathfrak{p}$ in R; and since $\mathfrak{P}^i$ has order q^{m-i} where $q = N(\mathfrak{p})$, for each i we can choose an element in $\mathfrak{P}^{i-1}$ but not in $\mathfrak{P}^i$. □

In particular, given a $\mathbb{Z}$-module basis of an ideal $\mathfrak{a}$, we can easily compute the unique $n \in \mathbb{N}$ such that $\mathfrak{a} \cap \mathbb{Z} = n\mathbb{Z}$; then we can just list the elements of the finite quotient $\mathfrak{a} \bmod n\mathcal{O}_K$ until we find a y such that $\mathfrak{a} = \langle n, y \rangle$. This representation of $\mathfrak{a}$ isn't as canonical as a Hermite-form $\mathbb{Z}$-basis, but it is much shorter if K has large degree, and hence often more convenient to work with in practice. In PARI/GP this is implemented as `idealtwoelt`; in Sage it is `gens_two()`.

Remark. Note that we have a lot of different notions of "generating set" here, for a number field of degree d and a nonzero ideal $\mathfrak{a} \trianglelefteq \mathcal{O}_K$:

- bases of $\mathcal{O}_K$, or of $\mathfrak{a}$, as a $\mathbb{Z}$-module (minimal size d, canonical choice from Hermite form)
- generating sets of $\mathcal{O}_K$ as a $\mathbb{Z}$-algebra (minimal size can be anything from 1 up to $1 + \log_2(d)$, no canonical choice)
- generating sets of $\mathfrak{a}$ as a $\mathcal{O}_K$-module (minimal size either 1 or 2, no canonical choice).

7.4 The unit group

Closely bound up with the arithmetic of ideals in $\mathcal{O}_K$ is understanding the group $\mathcal{O}_K^\times$, the *unit group* of K.

Example. The golden ratio $\phi = \frac{1+\sqrt{5}}{2}$ is a unit in $\mathbb{Q}(\sqrt{5})$, since $\phi \in \mathcal{O}_K$ and $1/\phi = \phi - 1 \in \mathcal{O}_K$.

Since the matrix of multiplication by x in any basis of $\mathcal{O}_K$, is a matrix of integers, it is invertible iff it has determinant ± 1. So $\mathcal{O}_K^\times$ is precisely the elements of $\mathcal{O}_K$ with norm ± 1.

There is a link between units and field embeddings into $\mathbb{C}$. The embeddings $K \hookrightarrow \mathbb{C}$ come in two kinds: real embeddings, and non-real complex embeddings, which come in pairs interchanged by complex conjugation.

Definition. *We say K has* signature (r_1, r_2) *if it has r_1 real embeddings and r_2 conjugate pairs of complex embeddings (so $d = r_1 + 2r_2$).*

Theorem (Dirichlet)**.** *Suppose K has signature (r_1, r_2). Then*

$$\mathcal{O}_K^\times \cong W_K \times \mathbb{Z}^{r_1+r_2-1}$$

where W_K is a finite cyclic group (containing ± 1).

In particular $\mathcal{O}_K^\times$ is finite if and only if $(r_1, r_2) = (1, 0)$ or $(0, 1)$, i.e. K is $\mathbb{Q}$ or an imaginary quadratic field.

7.4.1 Computing W_K

The group W_K is obviously computable. Clearly if K embeds in $\mathbb{R}$ then $W_K = \{\pm 1\}$. If K has no real embedding, then it can be larger. However, since the degree of the minimal polynomial of an n-th root of unity is $\varphi(n) = |(\mathbb{Z}/n)^\times|$, which goes to ∞ with n. If W_K has order n, then $\varphi(n)$ divides d, so there are only finitely many possibilities to check. (This can be hugely speeded up using information about the discriminant, since a field containing a p-th root of unity must have discriminant divisible by p).

Example. If $[K : \mathbb{Q}] = 2$ and W_K has order n, then $\varphi(n)$ must be 1 or 2; so $n = 1, 2, 3, 4$ or 6 (exercise). Clearly n must be even since $-1 \in K$. Thus, if $K = \mathbb{Q}(\sqrt{d})$ with d squarefree, we have

$$W_K = \begin{cases} \{\pm 1, \pm i\} & \text{if } d = -1 \text{ (order 4)} \\ \{\pm 1, \frac{\pm 1 \pm \sqrt{-3}}{2}\} & \text{if } d = -3 \text{ (order 6)} \\ \{\pm 1\} & \text{otherwise (order 2).} \end{cases}$$

7.4.2 Computing the free part

It remains to find a list of units $u_1, \dots, u_m$, where $m = r_1 + r_2 - 1$, that are a $\mathbb{Z}$-basis for $\mathcal{O}_K^\times / W_K$.

Definition. *Let $\sigma_1, \dots, \sigma_{r_1}$ be the real embeddings, and $\tau_1, \bar{\tau}_1, \dots, \tau_{r_2}, \bar{\tau}_{r_2}$ the complex embeddings (in conjugate pairs). Then the* logarithmic embedding *is the map*

$$\mathcal{L} : \mathcal{O}_K^\times \to V = \mathbb{R}^{r_1+r_2},$$

$$u \mapsto \Big(\log|\sigma_1(u)|, \dots, \log|\sigma_{r_1}(u)|, 2\log|\tau_1(u)|, \dots, 2\log|\tau_{r_2}(u)| \Big).$$

This is a group homomorphism. Its kernel clearly contains W_K. Since we have

$$\sum_i \log|\sigma_i(u)| + 2\sum_j \log|\tau_j(u)| = \log|N(x)| = \log 1 = 0,$$

the image is contained in the subspace $V_0 = \{(x_1, \dots, x_{r_1+r_2}) : \sum x_k = 0\}$. The proof of Dirichlet's theorem relies on showing that:

- the kernel of $\mathcal{L}$ is exactly W_K, and W_K is finite;

- the image of W_K is a *lattice* in V_0 (i.e. a discrete subgroup which spans V_0 over $\mathbb{R}$), and thus is necessarily isomorphic to $\mathbb{Z}^{\dim V_0}$.

Definition. *The* regulator *of K, Reg_K, is the volume of the quotient $V_0/\mathcal{L}(\mathcal{O}_K)$ (a positive real number).*

Concretely, if $u_1, \dots, u_m$ are a basis of $\mathcal{O}_K^\times/W_K$, then the matrix with rows $\mathcal{L}(u_i)$ has size $m \times (m+1)$ and its columns sum to 0. So if we form an $m \times m$ square matrix by deleting one column, the absolute value of the determinant doesn't depend on which column we threw away. This absolute value is the regulator of K.

Example. If K is a real quadratic field (or a cubic field of signature $(1,1)$), then there is a unique $u \in \mathcal{O}_K^\times$, the *fundamental unit* of K, such that $\sigma_1(u) > 1$ and $\sigma_1(u)$ is as small as possible subject to this. Then $\langle u \rangle = \mathcal{O}_K^\times / \pm 1$, and $\mathrm{Reg}_K = \log u_K$.

(Fundamental units can be surprisingly large; e.g. for $K = \mathbb{Q}(\sqrt{46})$, the group $\mathcal{O}_K^\times$ is generated by -1 and $24335 + 3588\sqrt{46}$.)

7.5 The class group

We saw a moment ago that sometimes $\mathcal{O}_K$ can fail to be a PID (or UFD). It turns out we can quantify *how badly* unique factorization fails. For this we need an algebraic gadget.

Definition. *A* fractional ideal *of $\mathcal{O}_K$ is a finitely-generated*[1] *$\mathcal{O}_K$-submodule of K; equivalently, it's a submodule $\frac{1}{N}\mathfrak{a}$, for a genuine ideal $\mathfrak{a} \trianglelefteq \mathcal{O}_K$ and $N \in \mathbb{Z}_{\geqslant 1}$.*

It turns out that fractional ideals form a group (the existence of inverses is the non-obvious part); and this group is a free abelian group of countably infinite rank, with basis given by the set of (non-zero) prime ideals. Moreover, the principal fractional ideals form a subgroup.

Definition. *The* class group *is the quotient*

$$\mathrm{Cl}(K) = (\textit{fractional ideals}) / (\textit{principal fractional ideals}).$$

Its elements are called ideal classes. *The* class number *h_K is the order of* $\mathrm{Cl}(K)$.

The ring $\mathcal{O}_K$ is a PID (or UFD) if and only if $\mathrm{Cl}(K) = \{1\}$, so h_K measures the extent of the failure of unique factorization.

Theorem (Minkowski). *Every ideal class contains an integral ideal of norm at most M_K, where*

$$M_K = \sqrt{|\Delta_K|}\left(\tfrac{4}{\pi}\right)^{r_2} \tfrac{d!}{d^d}.$$

In particular, h_K is finite.

[1] Either as an $\mathcal{O}_K$-module or a $\mathbb{Z}$-module – these are equivalent since $\mathcal{O}_K$ is itself finitely generated over $\mathbb{Z}$.

Example. For $K = \mathbb{Q}(\sqrt{-6})$, we have $|\Delta_K| = 24$, $d = 2$ and $r_2 = 1$, so $M_K = \frac{4\sqrt{6}}{\pi} \sim 3.1$. Thus every ideal class contains an ideal of norm 1, 2 or 3, i.e. it contains one of $\langle 1\rangle$, $\mathfrak{P} = \langle 2, \sqrt{-6}\rangle$, or $\mathfrak{Q} = \langle 3, \sqrt{-6}\rangle$.

We have $\mathfrak{P}^2 = \langle 2\rangle$ and similarly $\mathfrak{Q}^2 = \langle 3\rangle$ so both $\mathfrak{P}$ and $\mathfrak{Q}$ have order 2 in the class group. Since neither 2 or 3 is of the form $x^2 + 6y^2$, no element has norm 2 or 3 and hence these ideals must be non-principal.

Since the class of $\mathfrak{P}\mathfrak{Q}$ cannot be equal to that of $\mathfrak{P}$ or $\mathfrak{Q}$, and no other ideals lie below the bound, we conclude $\mathfrak{P}\mathfrak{Q}$ must be principal (we can check that $\mathfrak{P}\mathfrak{Q} = \langle\sqrt{-6}\rangle$, but we don't need to). Hence $\mathrm{Cl}(K) = C_2$, generated by $\mathfrak{P}$.

Remark. It's known that the class number of $\mathbb{Q}(\sqrt{-d})$ goes to ∞ with d, and there are exactly nine imaginary quadratic fields of class number 1, namely $d = \{1, 2, 3, 7, 11, 19, 43, 67, 163\}$. Those of you who went to Sarah's "introduction to number theory" talk will know that this has something to do with $e^{\pi\sqrt{163}}$, $e^{\pi\sqrt{67}}$, etc being spookily close to integers.

The situation for real quadratic fields is very different and poorly understood; it's conjectured that $\mathbb{Q}(\sqrt{p})$ has class number 1 infinitely often, but this is an open problem. (We don't even know if there are infinitely many number fields, of whatever degree and signature, having class number 1.)

7.5.1 Two pretty applications

Theorem. *The only integers x, y satisfying $x^3 - y^2 = 13$ are $(7, \pm 70)$.*

Proof. We write the equation as $x^3 = (y - \sqrt{-13})(y + \sqrt{-13})$.

Messing around with congruences, one can check that x must be odd, y must be even, and 13 does not divide x. Hence the ideals $\langle y - \sqrt{-13}\rangle$ and $\langle y + \sqrt{-13}\rangle$ have no common factor in $\mathbb{Z}[\sqrt{-13}]$.

Since these ideals are coprime and their product is $\langle x\rangle^3$, we must have

$$\langle y + \sqrt{-5}\rangle = \mathfrak{a}^3, \qquad \langle y - \sqrt{-5}\rangle = \mathfrak{b}^3$$

for some ideals $\mathfrak{a}, \mathfrak{b}$. But the class group has order 2, and $\mathfrak{a}^3$ and $\mathfrak{b}^3$ are principal; so $\mathfrak{a}$ and $\mathfrak{b}$ are themselves principal, say $\mathfrak{a} = \langle\alpha\rangle$, $\mathfrak{b} = \langle\beta\rangle$. Thus $y + \sqrt{-5} = \pm\alpha^3$, and replacing α by $-\alpha$, we can assume $y + \sqrt{-13} = \alpha^3$.

If $\alpha = m + n\sqrt{-13}$, then we get

$$m^3 - 39mn^2 = y, \qquad 3m^2n - 13n^3 = 1.$$

The second equation is $n(3m^2 - 13n^2) = 1$, so $n = \pm 1$; and hence $3m^2 - 13 = \pm 1$, meaning $3m^2 = 14$ (impossible) or $3m^2 = 12$, so $m = \pm 2$. Since $(-2 - \sqrt{-13})^3 = 70 + \sqrt{13}$, we get the above solution. □

Remark. The general result is that for any nonzero k, the equation $x^3 - y^2 = k$ has only finitely many integral solutions. The above method allows us to find them all whenever $k > 0$ and $\mathbb{Q}(\sqrt{-k})$ has class number coprime to 3.

Theorem (Kummer: Fermat's Last Theorem for regular primes)**.** *Let $p > 2$ be prime and suppose that $p \nmid h_K$, where $K = \mathbb{Q}(e^{2\pi i/p})$. Then $x^p + y^p = z^p$ has no integer solutions.*

(I'm not going to prove this here – we have all the conceptual tools needed, but the calculations are still quite hard. For a nearly-complete proof see Stewart and Tall. The missing details are in Washington *Introduction to cyclotomic fields*.)

7.6 The key computational problems

The key problems we have to solve, in order to compute in a number field, are the following.

- COMPUTING THE UNITS: find a generating set for $\mathcal{O}_K^\times / W_K$, and an algorithm for expressing an arbitrary element of $\mathcal{O}_K^\times$ in terms of these generators.
- PRINCIPALITY TESTING FOR IDEALS: given an ideal $\mathfrak{a}$ (via a Hermite-form $\mathbb{Z}$-basis), determine if it is principal, and if so, find a generator.

For imaginary quadratic fields, both are easy: $\mathcal{O}_K^\times / W_K$ is trivial; and given an ideal $\mathfrak{a}$, we can list the elements of norm equal to $\pm N(\mathfrak{a})$ and test each one to see if it fits.

For general fields the second problem is closely linked to the first one. We can't list all elements of norm $\pm n$ in $\mathcal{O}_K$, for a given n, because it's probably an infinite set. But there are finitely many *orbits* of norm $\pm n$ up to multiplication by $\mathcal{O}_K^\times$ (one for each principal ideal of norm n), and if we can enumerate those, then we're done. So these problems are closely linked.

7.6.1 Real quadratic fields

Let's assume K is a real quadratic field. Then $\mathcal{O}_K^\times = \pm 1 \times \langle u \rangle$ for some infinite-order u. Replacing u with $\pm u^{\pm 1}$ we can suppose that $u > 1$ (for the "obvious" embedding $\sigma : K \hookrightarrow \mathbb{R}$); then u is uniquely determined – we call it *the* fundamental unit u_K of K. It is characterized as the unit such that $\sigma(u)$ is > 1, but as small as possible.

Proposition. *We have $u_K = \frac{1}{2}(a + b\sqrt{D})$ for some integers $a, b > 0$ satisfying $a^2 - Db^2 = \pm 4$. If (a', b') is any other solution to this equation with $a', b' > 0$, then $a' > a$.*

Proof. If $u_K = \frac{1}{2}(a + b\sqrt{D})$, then

$$a = \frac{1}{2}(a + b\sqrt{D}) + \frac{1}{2}(a - b\sqrt{D}) = u_K + \tfrac{N(u_K)}{u_K} = u_K \pm \tfrac{1}{u_K},$$

and similarly $a' = u' \pm \frac{1}{u'}$ where $u' = \frac{a' + b'\sqrt{D}}{2}$.

We must have $u' = u^k$ for some $k > 1$, so $u' > u$. The functions $x \mapsto x + \frac{1}{x}$ and $x \mapsto x - \frac{1}{x}$ are both increasing on $(1, \infty)$; so if u_K and u have the same norm, or if u' has norm $+1$ and u has norm -1, then we are done. However, if u has norm $+1$ and u' has norm -1, we have a contradiction since u' is a power of u. □

So we can simply loop through positive integers a and compute whether $a^2 \pm 4$ is D times a square; the first such a we find will give us the fundamental unit. (Of course, if D is not 1 mod 4, it suffices to check even values of a.)

Remark. Sometimes u_K has norm $+1$ and sometimes it has norm -1. When $D = 1 \bmod 4$, it sometimes lies in $\mathbb{Z}[\sqrt{D}] \subset \mathcal{O}_K$ and sometimes not. There are no simple criteria for which case will occur.

Corollary. *Let $n \in \mathbb{Z}_{\geqslant 1}$. Then every $\mathcal{O}_K^\times$-orbit of elements of $\mathcal{O}_K$ of norm $\pm n$ has a representative $a + b\sqrt{d}$ with*

$$|a| \leqslant (nu_K)^{1/2}.$$

Proof. Consider the line in $\mathbb{R}^2$ defined by $V_n = \{(x, y) : x + y = \log n\}$. Then the point $P_0 = (\frac{1}{2}\log n, \frac{1}{2}\log n)$ is in V_n.

If $\beta \in \mathcal{O}_K$ has norm n, then $\mathcal{L}(\beta)$ is also in V_n. Hence the elements $\{\mathcal{L}(u\beta) : u \in \mathcal{O}_K^\times\}$ are equally spaced along V_n with common difference $(\log u_K, -\log u_K)$. If we take the closest such point P to P_0, it must be of the form $P_0 + (\lambda, -\lambda)$ with $|\lambda| < \frac{1}{2}\log u_K$. Thus its coordinates (x, y) satisfy $\max(|x|, |y|) \leqslant \frac{1}{2}\log n + \frac{1}{2}u_K$. Exponentiating, we get

$$\max(|a + b\sqrt{d}|, |a - b\sqrt{d}|) \leqslant (nu_K)^{1/2}$$

and the bound follows easily from this. □

Example. For $K = \mathbb{Q}(\sqrt{10})$, the equation $x^2 - 10y^2 = \pm 1$ has no solutions with $x = 1$ or $x = 2$, and $(3, 1)$ is a solution; so the fundamental unit is $3 + \sqrt{10}$ of norm -1.

The Minkowski bound is $\sqrt{10} < 4$, so every ideal class has a representative of norm 1, 2 or 3. From Dedekind–Kummer, we have

$$\langle 2 \rangle = \langle 2, \sqrt{10} \rangle^2, \qquad \langle 3 \rangle = \langle 3, 1 + \sqrt{10} \rangle \langle 3, 2 + \sqrt{10} \rangle.$$

There is an element of norm 6, namely $2 + \sqrt{10}$; so either all three ideals above are principal and the class number is trivial, or the class number is 2 and all three lie in the same non-principal ideal class.

To determine which, we need to find whether there are elements of norm ± 2. The bound above tells us that it suffices to check a, b with $\max(|a + b\sqrt{d}|, |a - b\sqrt{d}|) \leqslant \sqrt{2u_K} < 4$, hence $|a| < 4$ (and we can WLOG suppose $a \geqslant 0$). But the equations $0 - 10d^2 = \pm 2$, $1 - 10d^2 = \pm 2, \dots, 3 - 10d^2 = \pm 2$ have no solutions. Hence $\langle 2, \sqrt{10} \rangle$ is not principal and the class number is 2.

7.6.2 The general case

With a great deal more care, these ideas generalise to any number field.

Units

A rather useful result due to Remak (1932) says that there is a universal lower bound ϵ such that $\mathrm{Reg}_K > \epsilon$ for all number fields K (in fact we can take $\epsilon = 0.2$).[2]

Why is this useful? If we have some random list of units $v_1, \dots, v_m$, we might want to know if they are linearly independent in $\mathcal{O}_K^\times / W_K$. If we could compute with perfect precision we could just check if the matrix of their images under $\mathcal{L}$ is invertible. But we can only *approximately* compute the determinant of this matrix, so we might run into trouble if the determinant is very small.

However, the determinant has to be an integer multiple of Reg_K, and Reg_K is bounded below by Remak's result. So the determinant is either zero, or at least 0.2, and we can distinguish these possibilities with a finite amount of precision. Hence we can check if our units generate a finite-index subgroup, and give an upper bound for this index if so.

Similarly, once we've arranged that our v's generate a finite index subgroup, then given any $u \in \mathcal{O}_K^\times$, we can use interval arithmetic and RREF to compute approximations to the basis coefficients for $\mathcal{L}(u)$ in the basis $\mathcal{L}(v_1), \dots, \mathcal{L}(v_k)$. Since we know these are rational numbers whose denominators are bounded, this approximate information suffices to determine if u is in the subgroup generated by the v_i, and if not, determine a basis of the larger subgroup generated by u and the v_i.

Computing elements by norm

We're going to suppose we have the following:

- generators for a subgroup U of finite index in $\mathcal{O}_K^\times / W_K$;
- an integer $n \geqslant 1$.

We want to find representatives for the orbits in $\{x \in \mathcal{O}_K : N(x) = \pm n\}$ up to multiplication by U. If we fix a point $w \in V_n$, then the set

$$\{w + a_1\mathcal{L}(v_1) + \cdots + a_m\mathcal{L}(v_m) : 0 \leqslant a_i < 1\}$$

hits every orbit of V_n under $\mathcal{L}(U)$ exactly once; and it's obviously bounded. Hence its preimage in $\mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ is also bounded, and thus contains finitely many elements of $\mathcal{O}_K$, which we can enumerate explicitly.

If we do this with $n = 1$, then this allows us to find a finite collection of units which fills up the quotient $\mathcal{O}_K^\times / U$, so we can compute a basis of $\mathcal{O}_K^\times / W_K$. If we do this for some $n > 1$, then it will tell us all the principal lideals of norm n. Again, the fiddly thing here is the "precision management" issues caused by not being able to compute exactly in $\mathbb{R}$ or $\mathbb{C}$.

[2] There are other, much better bounds for specific classes of number fields, depending on the signature etc.

The end result

Eventually, we get an algorithm which (starting from a defining polynomial for K) will do the following, in finitely many steps:

- compute the ring of integers $\mathcal{O}_K$,
- find generators for $\mathcal{O}_K^\times$,
- determine the class number h_K and a set of ideals whose ideal classes generate Cl_K.

There are various computer programs which implement this, including Sage, PARI, and Magma.

8 Some hints at class field theory

8.1 Capitulating ideals

If we fix a number field K, then some ideals of $\mathcal{O}_K$ are principal, and if $h_K > 1$, then some aren't. But what if we change K?

Example. Let $K = \mathbb{Q}(\sqrt{-46})$ and let $\mathfrak{p} = \langle 5, 2+\sqrt{-46}\rangle$. Then $\mathfrak{p}$ is non-principal (and generates Cl_K, which is cyclic of order 4).

Let $L = K(\beta)$, where β is a root of the polynomial

$$x^4 + (2+\sqrt{-46})x^3 + (-23+\sqrt{-46})x^2 + (-10-5\sqrt{-46})x + (21-2\sqrt{-46}).$$

Then L is a Galois extension of K of degree 4, and the ideal $\mathfrak{p}\mathcal{O}_L$ turns out to be principal, generated by β. So $\mathfrak{p}$ *capitulates* (becomes principal) in L; as $\mathfrak{p}$ generates the class group, in fact every ideal of K capitulates in L. On the other hand, the class number of L is 3, so not every ideal of L is principal.

Kummer proved in 1840 that every ideal of a number field capitulates in some extension. But which extension? Is there somehow a "best" one? If we keep on going, do we always end up at a field with class number 1? These questions are addressed by *class field theory*, a beautiful and deep area which was developed in the first half of the 20th century, from Hilbert to Artin.

8.2 The Hilbert class field

It turns out we can always make ideals capitulate in a very specific kind of extension.

Definition (Unramified extensions)**.** *Let L/K be number fields and $\mathfrak{p}$ a prime of K. We say L/K is* unramified at $\mathfrak{p}$ *if $\mathfrak{p}\mathcal{O}_L$ is a product of distinct primes of L.*

We say L/K is unramified at ∞ *if every embedding $K \hookrightarrow \mathbb{R}$ extends to an embedding $L \hookrightarrow \mathbb{R}$ (rather than a non-real complex embedding). [This is vacuously true if K has no real embeddings.]*

If L/K is unramified at ∞ and every finite prime, and is Galois with abelian Galois group, we say it is an unramified abelian extension*.*

One of the great milestones of number theory is the following theorem, conjectured by Hilbert around 1900, and proved thirty years later by Artin and Furtwangler:

Theorem. *For any number field K, there is a unique largest unramified abelian extension of K, the* Hilbert class field *H_K, which contains every unramified abelian extension as a subfield. Moreover:*

- *there is a canonical isomorphism* $\mathrm{Gal}(H_K/K) \cong \mathrm{Cl}_K$*, so in particular* $[H_K : K] = h_K$*;*
- *every ideal of* $\mathcal{O}_K$ *capitulates in* H_K*.*

Remark. If L is any finite extension of K, then H_L contains H_K. As a corollary if this, if K embeds into a number field L with class number 1, then L contains the whole sequence of fields $\left(K, H_K, H_{H_K}, H_{H_{H_K}}, \dots\right)$ – the *class field tower* of K. In particular, K embeds into a field with class number 1 iff the class field tower eventually terminates.

In my example, K has degree 2, $H_K = K(\beta)$ degree 8, H_{H_K} degree 24, and we stop there; that field has class number 1. However, there do exist fields K for which the class field tower is infinite. The first example (due to Golod and Shafarevich from the 1960's) is that the class field tower is infinite if $K = \mathbb{Q}(\sqrt{-d})$ for some $d > 0$ with at least six distinct prime factors. So such a K cannot be embedded into any field with class number 1.

8.3 Ray class groups

For simplicity we assume henceforth that K has no real embeddings.

Class field theory doesn't only see unramified extensions; it turns out we can see *all* abelian extensions using a generalisation of the class group.

Definition. *For any ideal* $\mathfrak{m}$*, the fractional ideals of K coprime to* $\mathfrak{m}$ *form a group, and the principal ideals admitting a generator* x *with* $x = 1 \bmod \mathfrak{m}$ *are a subgroup. The quotient is called the* ray class group modulo $\mathfrak{m}$*, denoted* $\mathrm{Cl}_K(\mathfrak{m})$*.*

There is a natural map $\mathrm{Cl}_K(\mathfrak{m}) \to \mathrm{Cl}_K$, and this turns out to be surjective (this is not totally obvious – it relies on the fact that every ideal class has a representative coprime to $\mathfrak{m}$). This fits into a short exact sequence

$$1 \to \frac{(\mathcal{O}_K/\mathfrak{m})^\times}{\text{image } \mathcal{O}_K^\times} \to \mathrm{Cl}_K(\mathfrak{m}) \to \mathrm{Cl}_K \to 1,$$

so in particular $\mathrm{Cl}_K(\mathfrak{m})$ is always finite, and computable without too much difficulty once we know Cl_K and $\mathcal{O}_K^\times$.

On the field-extension side, we can attach to each abelian extension L/K an ideal $\mathfrak{f}_{L/K}$ of $\mathcal{O}_K$, the *conductor*, which measures the bad primes and how bad they are. (The primes dividing $\mathfrak{f}_{L/K}$ are exactly the primes that ramify in L/K.) Then we get the following theorem.

Theorem. *For any* $\mathfrak{m}$*, there exists a unique abelian extension* $K(\mathfrak{m})/K$ *which has conductor dividing* $\mathfrak{m}$ *and contains every other abelian extension of conductor dividing* $\mathfrak{m}$ *as a subfield. This field is called the* ray class field of K modulo $\mathfrak{m}$*. We have* $\mathrm{Gal}(K(\mathfrak{m})/K) \cong \mathrm{Cl}_K(\mathfrak{m})$*; and every ideal* $\mathfrak{a}$ *of K coprime to* $\mathfrak{m}$ *will "*$\mathfrak{m}$*-capitulate" in* $K(\mathfrak{m})$*, i.e.* $\mathfrak{a}\mathcal{O}_{K(\mathfrak{m})}$ *has a generator that is 1 modulo* $\mathfrak{m}\mathcal{O}_{K(\mathfrak{m})}$*.*

Since every abelian extension has some conductor, it must be contained in $K(\mathfrak{m})$ for some $\mathfrak{m}$ – hence, if we understand ray class fields, we understand all abelian extensions.

8.4 Computing class fields

So, here's the key question: can we compute the Hilbert class field of a given number field?

More generally, since $\mathrm{Gal}(H_K/K) \cong \mathrm{Cl}_K$, we get an inclusion-reversing bijection between subgroups $C \subseteq \mathrm{Cl}_K$ and unramified abelian extensions of K (and similarly for ray class fields). So we can split up our problem by writing Cl_K as a product of cyclic factors (of prime power order), and for each one of those, looking for the associated cyclic unramified extension.

8.4.1 The 2-part

First we consider the following simple case: can we find all unramified *quadratic* extensions of K?

Any quadratic extension has the form $K(\sqrt{\alpha})$ for some $\alpha \in K^\times$. Of course, the extension $K(\sqrt{\alpha})$ depends only on the class of $\alpha \in K^\times/(K^\times)^2$. So, for which classes in this quotient does it happen that $K(\sqrt{\alpha})$ is unramified? We have the following criterion:

Lemma. *If the extension $K(\sqrt{\alpha})/K$ is unramified at a prime $\mathfrak{p}$, then the power of $\mathfrak{p}$ dividing the fractional ideal $\langle\alpha\rangle$ is even. If $\mathfrak{p}$ does not lie above 2, the converse is true.* □

Definition. *We define the* two-Selmer group $\mathrm{Sel}_2(K)$ *to be the group of classes $[\alpha]$ in $K^\times/K^{\times 2}$ such that for every prime $\mathfrak{p}$, the power of $\mathfrak{p}$ dividing $\langle\alpha\rangle$ is even.*

Proposition. *The group* $\mathrm{Sel}_2(K)$ *is finite.*

Proof. Let α represent a class $[\alpha] \in \mathrm{Sel}_2(K)$. Then every prime ideal in the factorization of $\langle\alpha\rangle$ appears to an even power, so there is a unique ideal $\mathfrak{a}$ such that $\langle\alpha\rangle = \mathfrak{a}^2$. If we replace α with $\alpha\lambda^2$ for some λ, then we replace $\mathfrak{a}$ with $\mathfrak{a}\langle\lambda\rangle$, so the ideal class of $\mathfrak{a}$ depends only on $[\alpha]$. This defines a map

$$\mathrm{Sel}_2(K) \to \mathrm{Cl}_K,$$

which is obviously a group homomorphism. Its image is exactly the subgroup $\mathrm{Cl}_K[2]$ of elements of order 2 (since if $\mathfrak{a}^2$ is principal, any generator of $\mathfrak{a}^2$ defines a class in the Selmer group).

If $[\alpha]$ maps to the identity, then $\langle\alpha\rangle = \langle\beta\rangle^2$ for some β, and hence $u = \alpha/\beta^2$ is a unit. Thus $[\alpha] = [u]$ lies in the image of $\mathcal{O}_K^\times$ in $\mathrm{Sel}_2(K)$, and this image is clearly $\mathcal{O}_K^\times/\mathcal{O}_K^{\times 2}$, which is a finite group since $\mathcal{O}_K^\times$ is finitely generated. Hence we have a short exact sequence

$$1 \to \mathcal{O}_K^\times/\mathcal{O}_K^{\times 2} \to \mathrm{Sel}_2(K) \to \mathrm{Cl}_K[2] \to 1,$$

and since the first and last terms are finite, so is the middle term. □

This proof clearly gives us an algorithm: assuming we know Cl_K and $\mathcal{O}_K^\times$, we can write down a list of elements of $K^\times$ representing the elements of $\mathrm{Sel}_2(K)$. The extensions $K(\sqrt{\alpha})$ for $[\alpha] \in \mathrm{Sel}_2(K)$ aren't always unramified (they can be ramified at primes above 2, or at ∞), but every unramified quadratic extension will be in this list somewhere.

Remark. Actually we can easily check which extensions are unramified at ∞: the extension $K(\sqrt{\alpha})$ is unramified at ∞ iff $\sigma(\alpha) > 0$ for every real embedding σ. This condition cuts out a subgroup $\mathrm{Sel}_2^+(K)$ of $\mathrm{Sel}_2(K)$. However, identifying the classes unramified at primes above 2 is more fiddly.

Example. For the field $K = \mathbb{Q}(\sqrt{15})$, the fundamental unit of K is $u = 4 + \sqrt{15}$. So we get a subgroup $\mathcal{O}_K^\times/(\mathcal{O}_K^\times)^2 \cong C_2 \times C_2$ of $\mathrm{Sel}_2(K)$ with representatives $\pm 1, \pm u$. Of these, only 1 and u lie in $\mathrm{Sel}_2^+(K)$.

The class group is cyclic of order 2, generated by the prime $\langle 2, 1 + \sqrt{15}\rangle$ above 2, whose square is the principal ideal $\langle 2\rangle$. So $[2] \in K^\times/K^{\times 2}$ is an element of $\mathrm{Sel}_2(K)$, which clearly lies in $\mathrm{Sel}_2^+(K)$. Hence we get

$$\mathrm{Sel}_2^+(K) = \{[1], [u], [2], [2u]\}.$$

We compute that $K(\sqrt{2})/K$ is ramified at 2, and similarly for $K(\sqrt{u})/K$. However, $K(\sqrt{2u})$ is unramified at 2, so it is the Hilbert class field of K.

Remark. Note that the ideal $\langle 2, 1 + \sqrt{15}\rangle$ also capitulates in $K(\sqrt{2})/K$, even though this is not the Hilbert class field.

8.4.2 Generalisation: Kummer theory

Much the same argument works for degree n cyclic unramified extensions, as long as K contains an n-th root of unity. If so, any such extension has the form $K(\sqrt[n]{\alpha})$ for some $\alpha \in K^\times/K^{\times n}$ (a "Kummer extension").

As before, we can define the n-Selmer group $\mathrm{Sel}_n(K)$, consisting of elements $[\alpha] \in K^\times/K^{\times n}$ such that the fractional ideal $\langle \alpha \rangle$ is an n-th power. This sits in a short exact sequence

$$1 \to \mathcal{O}_K^\times/\mathcal{O}_K^{\times n} \to \mathrm{Sel}_n(K) \to \mathrm{Cl}_K[n] \to 1,$$

so it is finite and computable. Any unramified cyclic extension of degree n will be $K(\sqrt[n]{\alpha})$ for some $[\alpha] \in \mathrm{Sel}_n(K)$ (although we have to be careful that the Selmer group might also contain some "bad" extensions ramified at ∞ or at primes dividing n).

If we know that K has a cyclic unramified extension of degree n, but K doesn't have an n-th root of unity, we have to be a bit craftier:

Lemma. *If L/K is an unramified cyclic extension of degree n, then $L(\zeta_n)$ is an unramified cyclic extension of $K(\zeta_n)$ of degree dividing n.* □

So we can try listing *all* unramified cyclic extensions of $K(\zeta_n)$ of degree dividing n. Each of these contains finitely many degree n cyclic extensions of K, which can be enumerated explicitly using Galois theory. For each of those, we can check whether it's unramified (by computing something called a "relative discriminant", which is an ideal of K divisible precisely by the primes ramifying in the extension).

Example. Consider $K = \mathbb{Q}(\sqrt{-23})$. This has class number 3:

```
sage: K.<a> = QuadraticField(-23)
sage: K.class_number()
3
```

Can we find its Hilbert class field? To use Kummer theory we need to go up to the field $M = K(\zeta_3) = \mathbb{Q}(\sqrt{-23}, \sqrt{-3})$, which also has class number 3:

```
sage: M.<z3> = K.extension(x^2+x+1)
sage: M.class_number()
3
```

Since $[M : K]$ is coprime to 3, MH_K must be a cubic unramified extension of M; so in fact $MH_K = H_M$.

The 3-Selmer group of K is an $\mathbb{F}_3$-vector space of dimension 3:

```
sage: M.selmer_generators([], 3)
[-z3, (1/2*a + 3/2)*z3 + 3, -3*z3 - 1/2*a - 5/2]
```

The first two things in this list are units, generating $\mathcal{O}_M^\times/\mathcal{O}_M^{\times 3}$. The third element is a generator of $\mathfrak{a}^3$ where $\mathfrak{a}$ is a non-principal ideal:

```
sage: M.ideal( -3*z3 - 1/2*a - 5/2 ).factor()
(Fractional ideal (3, (-1/2*a - 1/2)*z3 + 1))^3
```

Each of the 26 nonzero elements in $\mathrm{Sel}_3(M)$ defines a cyclic extension of M, unramified except possibly at the primes above 3. Each extension occurs twice, since $K(\sqrt[3]{\alpha})$ and $K(\sqrt[3]{\alpha^2})$ are the same field; so we get 13 distinct extensions. Only one of these is an unramified extension, so this must be H_M:

```
sage: R.<x> = M[]
....: for b in M.selmer_group_iterator([], 3):
....:     if b == 1: continue
....:     N.<c> = M.extension(x^3 - b)
....:     if N.relative_discriminant() == 1 :
....:         HM = N
....:         print(HM)
....:         break
Number Field in c with defining polynomial x^3 - 3*a*z3 - 3/2*a - 25/2 over its base field
```

So $H_M = M\left(\sqrt[3]{\left(3\sqrt{-23}\right)\zeta_3 + \frac{3}{2}\sqrt{-23} + \frac{25}{2}}\right)$.

It remains to find H_K, among the subfields of H_M. We can list these using `HM.subfields()`. There are only four subfields of H_M which contain K, namely K, M, H_M, and one more which is a cubic extension of K; so this must be H_K.

```
sage: for A,_,_ in HM.subfields():
....:     if A.degree() == 6 and K.embeddings(A) != []:
....:         HK = A
....:         print(HK)
....:         break
....:
Number Field in c8 with defining polynomial x^6 - 3*x^5 + 5*x^4 - 5*x^3 + 5*x^2 - 3*x + 1
```

If we want a defining polynomial for H_K over K (not over $\mathbb{Q}$) we can use the `relativize` function:

```
sage: HK = _[0]
sage: HK2.<d> = HK.relativize(K.embeddings(HK)[0])
sage: HK2
Number Field in d with defining polynomial x^3 + (-1/2*a - 3/2)*x^2 + (1/2*a - 3/2)*x + 1 ov
```

Whew! That was quite an effort. Actually we could have done it in one step:

```
sage: HK.<d> = K.hilbert_class_field()
sage: HK
Number Field in d with defining polynomial x^3 - x^2 + 1 over its base field
```

This is the same field as before (non-obviously, since Sage's implementation has chosen a different defining polynomial).

Remark. The algorithm above, via Kummer extensions, is the only one which applies to any number field and is proved to always work. Unfortunately, it's very slow for large degrees. So there is a lot of interest in alternative methods.

If K is an imaginary quadratic field, there is a beautiful theory called "complex multiplication" which allows us to write down generators of Hilbert class fields using values of a complex-analytic function (the j-invariant).

The 12th of Hilbert's list of the 23 most important open problems in mathematics, from 1902, was to find a generalisation of this to any number field. This problem is still open, although there has been important progress recently thanks to Dasgupta and Kakde (who are giving a talk on this at ETH in July).

9 Interlude: public-key cryptography

9.1 Cryptographic algorithms

A *cryptographic protocol* is a technique for exchanging information securely. Typically, we imagine two people – Alice and Bob – who are trying to communicate, while a third participant, Eve, can see all their messages; and the goal is for Alice to send some information to Bob (or vice versa), without Eve being able to understand what it means. (We're assuming here that Eve is a passive participant; she can read everything, but can't steal away messages in transit and replace them with fakes, etc.)

So Alice needs some algorithm to take a message (*plaintext*) and produce some new message (*cyphertext*) which can be transmitted to Bob; and Bob needs an algorithm for the reverse process, reconstructing the plaintext from the cyphertext.

Alice and Bob can try to rely on making their entire system, the encryption and decryption algorithms, completely secret. However, this makes for an unwieldy and risky system; as soon as there's any danger that Eve has found out the secret (if Alice leaves her laptop on the bus or something), then Alice and Bob will need to throw away the whole system and develop a new set of protocols from scratch! So, typically, cryptographic protocols rely on an extra piece of data (the "key") which has to be kept secret. Even if Eve knows the algorithm that Alice and Bob are using, as long as she doesn't know the key, then their communication is secure.

9.1.1 Symmetric versus private-key

The most "obvious" form of key-based cryptography is what we call symmetric-key cryptography: Alice and Bob both need to know the same key, which has to be agreed on in advance and kept secret by both participants (a *shared secret*).

A **public-key cryptographic protocol** is designed to avoid the requirement for a key to be arranged in advance and kept secure. In a public-key system, Alice and Bob each have a "public key" and a "private key": as long as Bob knows Alice's public key, he can encrypt a message for her, but it can't be decrypted without her private key – as long as Alice keeps the private key secret, nobody (including Bob himself!) can decrypt the message.

Symmetric-key cryptography is a straighforward idea that has been in use for thousands of years. In contrast, public-key cryptography seems like it should be obviously impossible, and the first workable algorithm (the RSA algorithm) only dates back to the 1970's.

Even the best available public-key algorithms tend to be much slower to encrypt than symmetric-key systems. So it's common to use both together – Alice chooses a single-use "session key" for the symmetric-key algorithm, uses a separate public-key algorithm just to send Bob the session key, and then uses the symmetric-key algorithm (with the session key) to encrypt the actual message.

Remark. I think it's fair to say that, while symmetric-key cryptography is a hugely important problem, it's one that belongs squarely to computer science rather than mathematics. So I won't say any more about it here; I'll concentrate on public-key cryptography, which has more mathematical content.

9.1.2 The RSA algorithm

The RSA algorithm (named after the *second* set of people to discover it!) relies on computations in the unit group $G = (\mathbb{Z}/N)^\times$ for an integer N.

- Alice's public key is the pair (N, e), where e is an integer coprime to $\lambda(N) = \#G$.
- Her private key is $d = e^{-1} \bmod \lambda(N)$.
- If Bob wants to send a message to Alice, he writes his message as an element of G (or maybe a sequence of chunks which are elements of G).
- For each chunk $g = a \bmod N$, Bob computes $a^e \bmod N$ and sends it to Alice.
- Alice can then compute $(a^e)^d = a^{de} = a \bmod N$ and hence read the message.

The security of the algorithm relies on choosing N so that $\lambda(N)$ is hard to compute. If N is prime, then computing $\lambda(N) = N - 1$ is trivial; but if $N = pq$ is a product of 2 primes, then $\lambda(N) = (p-1)(q-1)$, so computing $\lambda(N)$ amounts to factoring N.

Thus the security of RSA relies on the fact that **factoring large composite numbers is difficult**.

Disadvantages: RSA is still occasionally used, but it has some disadvantages.

- To be secure against currently available computers, N needs to be fairly large (~ 3000 binary digits); this makes the encryption and decryption slow, and the message sizes large.
- There are lots of special tricks that can break the security; e.g. if one of p and q is too small, or p and q are too close together, or if you send the same message to two people with the same N and different e's, then Eve can deduce the private key. This means that implementing the algorithm in a really secure way is difficult to do.

9.1.3 Diffie–Hellmann key exchange

Remember we wanted a public-key protocol just as a way of sending a session key for another, symmetric-key protocol.

However, we can get away with even less: rather than Alice *choosing* an arbitrary key and sending it to Bob, it's enough for Alice and Bob to be able to settle on a key which they both know, and which Eve can't easily determine – even if the process doesn't allow Alice or Bob to specify in advance what that key will be. Once they have established this shared secret, they can use it as the session key for a symmetric-key protocol.

The **Diffie–Hellmann key exchange protocol** is one very effective way of establishing shared secrets. It relies on finding an abelian group G and an element $g \in G$ of finite, but large, order N; we suppose G, g, and N are public knowledge, and that group operations and equality testing in G are reasonably cheap to compute.

- Alice and Bob each choose an integer in the range $(0, N)$, say a for Alice and b for Bob; these are their private keys.
- They each compute the group elements $A = g^a$ and $B = g^b$ of G which they announce to the world; these are their public keys.
- Alice takes Bob's public key B, and her own private key a, and computes B^a.
- Bob likewise computes A^b.
- Since $(g^a)^b = (g^b)^a$, Alice and Bob have computed the same group element, giving them a shared secret.

Eve, on the other hand, knows the group elements g, g^a and g^b, and she wants to find out g^{ab}. This is called the *Diffie–Hellman problem* for G. If she can compute a given g^a (solving the *discrete logarithm problem*), then she can obviously solve the Diffie–Hellmann problem too. So we want to choose G in such a way that the DLP is hard.

(In principle it's possible that there might be groups where DLP is hard, but there is some clever trick for solving DH without going via DLP; but as far as I know there are no known examples, so these problems seem to be equally hard.)

9.2 Choosing the group

What G shall we take? If G has order N, then we can solve the DLP in G by brute force in $O(N)$ steps, so we want this to be out of range of practical computation. On the other hand, representing elements of G and computing with them has to take at least $O(\log N)$ steps, and usually is exactly this order. So we want G to land in the gap where $O(\log N)$ is reasonable, but $O(N)$ is beyond reach.

There are slightly better ways of solving DLP for a "generic" group, such as Pollard's ρ method, which will get you down from $O(N)$ to $O(\sqrt{N})$, but that is still unbridgeably far from $O(\log N)$.

Unit groups. Diffie and Hellmann originally took $G = (\mathbb{Z}/p)^\times$, where p is a large prime. Then G is cyclic, and we can choose g to be a generator of G (a primitive root mod p).

However, this protocol has a hidden flaw. There is a family of algorithms called *index calculus* which can be used to solve the discrete log problem in $(\mathbb{Z}/p)^\times$ in $O(e^{(\log p)^{1/2}})$, or even $O(e^{(\log p)^{1/3}})$, steps. This is not polynomial in $\log p$ but it's a lot less than exponential. One can also try using multiplicative groups $\mathbb{F}_q^\times$ where q is a prime power, but the index calculus attack can be used on these too, so they are also vulnerable.

Worse still, the expensive part of the computation depends only on p, and not on the specific element whose discrete logarithm we're trying to find. Since finding large primes is expensive, many Diffie–Hellmann implementations use the same few primes (of size about $2^{2^{10}}$); and it's widely believed that various governments' spy agencies might have done the pre-computation needed to break Diffie–Hellman for those primes, for all possible choices of key at once. We could increase the size of the primes used and be safe for a few more years, but that slows down the key-exchange computation.

9.3 Elliptic curves

As we'll see in the next chapter, elliptic curves over finite fields give rise to examples of abelian groups which seem to have no "special vulnerabilities". That is, there are no algorithms currently known for solving the DLP for these specific groups which are any faster than exponential in the key length.

The standard cryptographic protocol used by Web browsers today (TLS 1.3) uses elliptic curve Diffie–Hellmann as its key-exchange algorithm. So every time you type a password into a web site, you are using elliptic curves! There's a good answer if anyone ever challenges you to prove that advanced mathematics is useful in the real world.

10 Elliptic curves

10.1 Definitions

Let K be a field.

Definition. *A* Weierstrass equation *over K is an equation of the form*

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6$$

for some constants $a_1, \dots, a_6 \in K$.

The variety defined by a Weierstrass equation can be singular, or nonsingular. An elliptic curve *is a nonsingular variety defined by a Weierstrass equation.*

We say two elliptic curves are isomorphic *if they differ by a coordinate change*

$$x \mapsto \lambda^2 x + r, \qquad y \mapsto \lambda^3 y + sx + t$$

for some $\lambda \in K^\times$ and $r, s, t \in K$.

For short we can write "the elliptic curve $[a_1, \dots, a_6]$". Note that there isn't an a_5; the reason for this funny numbering is that if we do a coordinate change with $r = s = t = 0$ but λ arbitrary, then this coordinate change replaces a_i by $\lambda^i a_i$.

Lemma. *If K does not have characteristic 2 or 3, any elliptic curve is isomorphic to one having the form*

$$y^2 = x^3 + Ax + B,$$

(a short Weierstrass equation*), and the smoothness condition is $4A^3 + 27B^2 \neq 0$.*

10.2 The group law

Let E be an elliptic curve. If $K \subseteq L \subseteq \overline{K}$, let $E(L)$ be the set of points $(x, y) \in L^2$ lying on E, together with an extra point "at infinity" denoted $\mathcal{O}$.

Lemma. *Let L be a line in the affine plane. Then L intersects E at exactly three points (counted with multiplicity), unless L is a vertical line $x = c$ for some c, in which case there are exactly two intersections.*

In the third case we formally define the "third intersection point" to be $\mathcal{O}$. Then we have the following theorem:

Theorem. *There is a unique abelian group structure on $E(\overline{K})$ with the following properties:*

- *the point $\mathcal{O}$ is the identity;*
- *if L is any line and P, Q, R are its three intersection points with E, then $P + Q + R = 0$.*

Concretely, we can describe the group operation as follows. Let $P, Q \in E(\overline{K})$. Since we know $\mathcal{O}$ is the identity, we can assume neither P nor Q is $\mathcal{O}$.

- If $P \neq Q$, let L be the line through P and Q. If $P = Q$, let L be the tangent at P.
- Let R be the third intersection of L with E.
- Let L' be the vertical line through R. This intersects E at R and one other point, and we define this other point to be $P + Q$.

The tough thing is proving that this binary operation is *associative* (the other group axioms are relatively easy). For a proof, see the textbook by Silverman.

Explicit formulae

Suppose E is in short Weierstrass form. Given $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, with $x_1 \neq x_2$ (i.e. $P \neq \pm Q$), the equation of the line joining them is given by $y = \lambda x + \nu$, where

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \qquad \nu = \frac{y_1 x_2 - x_1 y_2}{x_2 - x_1}.$$

If $P = Q$, then we compute the tangent line using partial derivatives, and get

$$\lambda = \frac{3x_1^2 + A}{2y_1}, \qquad \nu = \frac{-x_1^3 + Ax_1 + 2B}{2y_1}.$$

In either case, we can find $L \cap E$ by solving

$$(\lambda x + c)^2 = x^3 + Ax + B, \Longleftrightarrow x^3 - \lambda^2 x + \cdots = 0.$$

The three roots have to sum to λ^2, giving x_3, and we can find y_3 from the equation of L. Hence

$$R = (x_3, y_3) = \left(\lambda^2 - x_1 - x_2, -\lambda^3 + \lambda(x_1 + x_2) - \nu\right).$$

Proposition. *(i) For any field L with $K \subseteq L \subseteq \overline{K}$, the subset $E(L) \subseteq E(\overline{K})$ is a subgroup.*

(ii) If the field operations in L are computable, then so is the group law on $E(L)$.

Proof. Clear from the formulae: if x_1, y_1, x_2, y_2 are in L, then so is everything else, and we can compute it by doing an explicit finite sequence of field operations. □

(This is also true for curves in long Weierstrass-form curves, even in the bad characteristics where short Weierstrass forms don't exist. It's the same proof, just with messier formulae.)

Corollary. *If $K = \mathbb{F}_q$ is a finite field, then $E(K)$ is a finite group in which the group operations are (efficiently) computable.*

10.3 Elliptic curves over finite fields

How big is $E(\mathbb{F}_q)$, for E an elliptic curve? It's not hard to see that

$$1 \leqslant \#E(\mathbb{F}_q) \leqslant 2q + 1.$$

The 1 is for $\mathcal{O}$; and for the points not at infinity, there are q possibilities for x and at most two y-values for each. However, it's actually always very close to the middle of this range.

If we wanted to make a list of points on E, we could list all $x \in \mathbb{F}_q$ and for each one, check if $f(x)$ is a square (giving 2 points on E) or a non-square (giving no points). Not forgetting the extra point at ∞, we have

$$\#E(\mathbb{F}_q) = q + 1 + \#\{x : f(x) \text{ nonzero square }\} - \#\{x : f(x) \text{ non-square }\}.$$

Since $\mathbb{F}_q^\times$ is a cyclic group of even order, exactly half its elements are squares. So we might expect these two sets to be about the same size, i.e. $\#E(\mathbb{F}_q)$ should be close to $q + 1$.

Exercise. Can you show that if $q = 3 \bmod 4$, and E has the form $y^2 = x^3 + Ax$ with $A \neq 0$, then $\#E(\mathbb{F}_q)$ is exctly $q + 1$? (Hint: find a bijection between the two sets of x-values.)

10.3.1 The Frobenius and Hasse's inequality

Let $t_q(E) = q + 1 - \#E(\mathbb{F}_q)$; we're expecting this number to be "small".

Theorem (Hasse's inequality). *Let E be an elliptic curve over $\mathbb{F}_q$. Then $|t_q(E)| \leqslant 2\sqrt{q}$.*

The proof of this theorem is rather deep, and we won't give the details here (see Silverman's book for a full account). It rests on computations with *endomorphisms* of elliptic curves – maps of algebraic varieties $E \to E$ over $\overline{\mathbb{F}_q}$ which are compatible with the group structure. The set $\mathrm{End}(E)$ is naturally a *ring*, using the group structure of E to define addition, and composition of endomorphisms for multiplication.

This works over any field, but if E is defined over $\mathbb{F}_q$, we have a special element of $\mathrm{End}(E)$, the *Frobenius* φ_q, defined by

$$(X, Y) \mapsto (X^q, Y^q).$$

Note that the points fixed by φ_q, i.e. the kernel of $1 - \varphi_q$, is exactly $E(\mathbb{F}_q)$.

The key point of Hasse's inequality is that the Frobenius *satisfies a quadratic polynomial* in $\mathrm{End}(E)$:

$$(\varphi_q)^2 - t_q\varphi_q + q = 0.$$

Hasse's inequality is then precisely the assertion that this polynomial can't have distinct real roots, which follows from computations involving degrees of isogenies.

A by-product of this argument is the following. If we go up from $\mathbb{F}_q$ to $\mathbb{F}_{q^k}$, we just replace φ_q by its k-th power, and we can compute the degree of $1 - (\varphi_q)^k$ if we know that of $1 - \varphi_q$. The result is the following:

Proposition. *Let α_q, β_q be the roots of the polynomial $X^2 - t_q X + q$. Then, for any field extension $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$, we have*

$$\#E(\mathbb{F}_{q^k}) = 1 + q^k - (\alpha_q)^k - (\beta_q)^k.$$

Remark. In cryptography it's common to use elliptic curves over large finite fields of characteristic 2, say $\mathbb{F}_{2^k}$ with $k \sim 100$. If we take an E which is actually defined over $\mathbb{F}_2$ (a "binary Koblitz curve"), then computing $t_2(E)$ is very quick, and then we can compute $t_{2^k}(E)$ easily from the previous proposition.

10.3.2 Computing $\#E(\mathbb{F}_q)$

We'd like to be able to compute $t_q(E)$, and hence $\#E(\mathbb{F}_q)$, in reasonable time, even if q is quite large. (For example, the cryptographic protocol used to encrypt Bitcoin transactions uses the curve $y^2 = x^3 + 7$ over $\mathbb{F}_p$, where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$, and we need to know $\#E(\mathbb{F}_p)$.)

Theorem (Schoof, 1985)**.** *There is an algorithm to compute $t = t_q(E)$, given q and the equation of E, whose complexity is polynomial in* $\log q$.

We'll now sketch how Schoof does this. We know that t fits into the quadratic polynomial

$$\phi^2 - t\phi + q = 0 \in \operatorname{End} E(\overline{\mathbb{F}}_q), \tag{†}$$

i.e. we have the relation

$$(X^{q^2}, Y^{q^2}) + q \cdot (X, Y) = t \cdot (X^q, Y^q) \quad \text{in} \quad \mathbb{F}_q[X,Y]/(Y^2 - X^3 - AX - B),$$

where the additions and multiplications are understood via the group law on E. In principle we could just solve for t from this, but this would be a terrible algorithm in practice, since the degrees of both sides in X grow polynomially with q.

Schoof's algorithm relies on taking (x, y) to be a *torsion point*: a point with $\ell \cdot (x, y) = 0$, for $\ell \neq p$ a small prime. This means that we are now trying to solve

$$(X^{q^2}, Y^{q^2}) + \bar{q}(X, Y) = \bar{t}(X^q, Y^q) \in \mathbb{F}_q[X,Y]/(Y^2 - X^3 - AX - B, \psi_\ell(X)),$$

where the bars denote reduction mod ℓ, and ψ_ℓ is the *ℓ-division polynomial* of E, vanishing precisely at the x-coordinates of ℓ-torsion points, whose degree is $(\ell^2 - 1)/2$. Now we never need to use polynomials of degree bigger than a fixed power of ℓ; and if ℓ is small (roughly log q in size), then this is a much more feasible computation. However, it only tells us t mod ℓ.

So Schoof makes a table of small primes $\ell_1, \ell_2, \ldots$ and computes $t_i = t \bmod \ell_i$ for each i; this determines $t \bmod L = \prod_i \ell_i$. If we take enough primes that $L > 4\sqrt{q}$, then Hasse's inequality gives a unique possibility for q; and the small primes are dense enough that the number of steps is still polynomial in $\log q$.

Example. For the Bitcoin curve, $4\sqrt{q} \sim 10^{39}$, which seems rather huge, but is actually less than $2 \times 3 \times 5 \times \cdots \times 103$ (the product of the first 27 primes).

[Schoof's algorithm would be a good project topic for students interested in algebraic geometry.]

Remark. There are other algorithms for computing t, which all start from $(\dagger)$. One interesting example is *Kedlaya's algorithm*, which computes the matrix of φ_q acting on a certain space of p-adic differential forms; the trace of this matrix is $t_q(E)$. There are also generalisations to curves of higher degree (despite the lack of a group operation).

10.3.3 Group structure

Having found the size of the group $E(\mathbb{F}_q)$, we can ask about its structure.

Theorem. *Let $K = \mathbb{F}_q$ be a finite field. Then $E(K)$ is isomorphic to $\mathbb{Z}/L \times \mathbb{Z}/LM$, for some integers L, M with $L \mid q-1$.*

The point is that $E(\overline{\mathbb{F}}_q)$ can't have more than n^2 elements of order n, for any n. So there can't be more than two cyclic groups in the decomposition. To see that L has to divide $q-1$, we use a construction called the *Weil pairing*, which is a skew-symmetric form

$$E(K)[n] \times E(K)[n] \to \mu_n(K) = \{x \in K^\times : x^n = 1\}.$$

Since $E(K)$ has L^2 points of order L, then $K^\times$ has to contain an L-th root of unity, implying that L has to divide $q-1$.

These constraints on L mean that it "usually" ends up being quite small, i.e. $E(K)$ is usually cyclic or close to it.

Exercise.

(i) Show that if $q = 2^k$ and $q-1$ is a prime, then every elliptic curve E over $\mathbb{F}_q$ has $E(\mathbb{F}_q)$ cyclic.

(ii) If $q = 17$, how many possible groups $\mathbb{Z}/L \times \mathbb{Z}/LM$ can occur which satisfy the conclusions of the last two theorems?

Having computed $\#E(\mathbb{F}_q)$ (e.g. via Schoof's algorithm), then we can compute $\gcd(\#E(\mathbb{F}_q), q-1)$. If we're lucky, this will be small enough to factorize into prime powers, and hence we can compute all the possibilities for L. For each candidate L, if it is not too large, we can compute how many L-torsion points there are on E by factorizing the L-division polynomial in $\mathbb{F}_q[X]$.

Remark. If $q-1$ and $\#E(\mathbb{F}_q)$ have a large common factor, then determining the structure of $E(\mathbb{F}_q)$ might be difficult. However, this case is less useful for cryptographic purposes anyway. Curves used in cryptography are generally chosen so that $\#E(\mathbb{F}_q)$ is either prime (e.g. the Bitcoin curve), or a small integer times a prime.

10.4 Elliptic curves over $\mathbb{Q}$

We might well want to consider elliptic curves over $\mathbb{Q}$. In fact it'll be useful to set up the theory more generally, and consider elliptic curves over number fields.

10.4.1 The Mordell–Weil theorem

Theorem (Mordell). *$E(K)$ is a finitely generated abelian group, so*

$$E(K) \cong \mathbb{Z}^r \times E(K)_{\text{tors}}$$

where $r \in \mathbb{Z}_{\geqslant 0}$ (the rank of E) and $E(K)_{\text{tors}}$ (the torsion subgroup) is a finite abelian group.

We'd like to compute: the torsion subgroup, the rank, and set of r points generating $E(K)$ modulo torsion.

Remark. Note the shift in perspective from the finite field case, where all the questions were *in principle* finite computations, and the problem was how to solve them efficiently when q was too large for a brute-force check. Now our fields are infinite, so the challenge is to show that the computations are even possible at all; we're not going to worry about how to do them efficiently!

10.4.2 Torsion and reduction

The torsion subgroup is "the easy bit". First, we note that for any given N we can compute all points of order N in $E(K)$, just by factorizing the N-division polynomial in $K[X]$. So we are safe as soon as we can find an upper bound for the order of points in $E(K)_{\text{tors}}$.

We can do this by *reduction modulo primes*. Let $\mathfrak{p} \lhd \mathcal{O}_K$; we suppose $\mathfrak{p}$ doesn't divide 2 or $4A^3 + 27B^2$. Then the reduction of the Weierstrass equation defines an ellpitic curve $\overline{E}$ over the finite field $k_{\mathfrak{p}} = \mathcal{O}_K/\mathfrak{p}$.

Given a non-zero point $P \in E(K)$, we can write it in projective form $(u : v : w)$ with u, v, w having denominators coprime to $\mathfrak{p}$, and not all divisible by $\mathfrak{p}$. Then $(\bar{u} : \bar{v} : \bar{w})$ define a point $\bar{P} \in \bar{E}(k_{\mathfrak{p}})$.

Proposition. *This map is a group homomorphism $E(K) \to \bar{E}(k_{\mathfrak{p}})$; and its kernel contains no nonzero torsion points.*

In particular, it follows that $\#E(K)_{\text{tors}}$ divides $\#\bar{E}(k_{\mathfrak{p}})$ (which we can compute). So we have an upper bound for its order.

Example. Consider $y^2 = x^3 + 4$ over $\mathbb{Q}$. The first prime not dividing $2(4A^3 + 27B^2)$ is $p = 5$; we find that $\bar{E}$ has 6 points, so the torsion subgroup of E must have order dividing 6. The 6-division polynomial has degree 19, but factoring a degree 19 polynomial is easy for a computer, and we find it has no roots except $x = 0$, which gives the points $(0, \pm 2)$. So $E(\mathbb{Q})_{\text{tors}} = \{\mathcal{O}, (0, 2), (0, -2)\} \cong C_3$.

(We could also be more sneaky and check $p = 7$ as well, and find that E mod 7 has only 3 points; so it would suffice to factor the 3-division polynomial instead, which is much smaller.)

Remark. Actually, for $K = \mathbb{Q}$, we always have $\#E(\mathbb{Q})_{\text{tors}} \leqslant 12$. This is a very deep theorem of Mazur. More generally, Merel showed that for any number field K there is a bound, depending only on K, for how big $E(K)_{\text{tors}}$ can be for an elliptic curve E/K. However, if we are given a specific E, then computing reduction mod p is easy and cheap, and usually gives much better bounds.

10.4.3 Computing non-torsion points

A significant stepping-stone towards computing $E(K)$ is the following result:

Theorem (Weak Mordell–Weil)**.** *For any integer N, the group $E(K)/NE(K)$ is finite.*

This is clearly a consequence of full Mordell–Weil. The standard proof of Mordell–Weil involves proving the weak form first, and then doing some extra work to deduce the full theorem from this.

However, it is also useful for computations: once we know that Mordell-Weil is true, it follows that

$$\frac{E(K)/NE(K)}{\text{image of } E(K)_{\text{tors}}} \cong (\mathbb{Z}/N\mathbb{Z})^r,$$

and any set of elements of $E(K)$ whose images are a minimal generating set of this quotient will actually generate a finite-index subgroup of $E(K)$.

So computing this quotient is the key to understanding the K-points of E. This is known as N**-descent**.

Let's suppose K is big enough that all the N-torsion points in $E(\overline{K})$ are actually in $E(K)$. Given a point $P \in E(K)$, we can choose a point $\tilde{P} \in E(\overline{K})$ with $N\tilde{P} = P$. We can then define a map

$$\lambda_P : \mathrm{Gal}(\overline{K}/K) \to E(K)[N], \qquad \lambda_P(\sigma) = \sigma(\tilde{P}) - \tilde{P}.$$

This doesn't depend on the choice of $\tilde{P}$ lifting P; and if P is actually in $NE(K)$, then we can choose $\tilde{P}$ to be in $E(K)$, so λ_P is the zero map. This construction defines an *injective* map

$$E(K)/NE(K) \hookrightarrow \mathrm{Hom}\left(\mathrm{Gal}(\overline{K}/K), E(K)[N]\right) \cong \mathrm{Hom}\left(\mathrm{Gal}(\overline{K}/K), \mathbb{Z}/N\mathbb{Z}\right)^{\oplus 2}$$

$$\cong \left[K^\times/(K^{\times N})\right]^{\oplus 2}.$$

This isn't a lot of help on the face of it, since the target is still infinite. However, one can check that the image lands in a certain subgroup of $K^\times/(K^{\times N})$ (defined by conditions on the prime factorisation):

Definition. *For S a finite set of primes, let $\mathrm{Sel}_N(K,S)$ be the group of classes $[\alpha] \in K^\times/K^{\times N}$ such that for all primes $\mathfrak{p} \notin S$, the power of $\mathfrak{p}$ dividing α is a multiple of N.*

If $S = \varnothing$, this is the N-Selmer group which we saw in chapter 8; and the arguments there extend to show that $\mathrm{Sel}_N(K,S)$ is finite (and computable) for any finite set S. For instance, if $K = \mathbb{Q}$, then $\mathrm{Sel}_2(\mathbb{Q},S) \cong (\mathbb{Z}/2)^{\#S+1}$, generated by ± 1 and the elements $[p]$ for $p \in S$.

Proposition. *The image of $E(K)/NE(K)$ is contained in $\mathrm{Sel}_N(K,S)^{\oplus 2}$, where S is the set of primes dividing $N\Delta(E)$.*

So we can embed $E(K)/NE(K)$ inside a finite set, which proves the weak Mordell–Weil theorem (at least under our rather strong hypotheses on K).

We can be a bit more concrete for $N = 2$. The assumption that all the 2-torsion points are defined over K means we can write E as $y^2 = (x - e_1)(x - e_2)(x - e_3)$, with $e_i \in K$. Then there is an injection

$$E(K)/2E(K) \to \mathrm{Sel}_2(K, S) \times \mathrm{Sel}_2(K, S),$$

defined by $(x, y) \mapsto ([x - e_1], [x - e_2])$ if $x \neq e_1, e_2$ (with various formulae for the special cases when (x, y) is a 2-torsion point).

Proposition. *Let $([b_1], [b_2]) \in \mathrm{Sel}_2(K, S)$. Then $([b_1], [b_2])$ is in the image of $E(K)/2E(K)$ if either*

- *it is one of the special elements $(1, 1)$, $(\frac{e_1-e_3}{e_1-e_2}, e_1 - e_2)$, $(e_2 - e_1, \frac{e_2-e_3}{e_2-e_1})$ (which are the image of $E(K)[2]$);*
- *or, the equations*

  $$b_1 z_1^2 - b_2 z_2^2 = e_2 - e_1, \qquad b_1 z_2^2 - b_1 b_2 z_3^2 = e_3 - e_1$$

  have a solution $(z_1, z_2, z_3) \in K^\times \times K^\times \times K$, in which case we can take

  $$P = (b_1 z_1^2 + e_1, b_1 b_2 z_1 z_2 z_3) \in E(K).$$

If we list all the pairs (b_1, b_2), then usually we will either spot an obvious solution, or there will be a cheap trick (using congruences modulo various primes) to show that no solutions exist. If this works, it gives us a finite set of points generating $E(K)/2E(K)$.

11 Group theory I: finitely-presented groups

11.1 Presentations of groups

How can we describe a group? One important description is by specifying a list of generators and a list of relations between them (a *presentation* of a group). We're interested in the case when the list of generators and the list of relations are both finite. This clearly doesn't always imply that G itself is finite!

Example. Let G be the group generated by 2 elements a, b with the relations $a^2 = b^2 = (ab)^2$. What does G look like? Is it finite?

The difficulty here is that *there cannot be a general algorithm* for such questions!

Theorem. *Each of the following problems is* undecidable*:*

- *Given a finite presentation $G = \langle g_1, \ldots, g_n \mid r_1, \ldots, r_m \rangle$, determine if G is the trivial group.*
- *Given a finite presentation, determine if G is finite.*
- *Given two finite presentations, determine whether the groups they describe are isomorphic.*

These are consequences of the *Novikov–Boone theorem* in group theory; see Chapter 12 of Rotman's book for more details. What "undecidable" means here is that there is no algorithm to solve the problem which is guaranteed to terminate in finitely many steps (on any valid input).

However, we can get away with something weaker:

Proposition. *For each of the problems above, there exists an algorithm which will terminate if and only if the groups are finite (and will run forever otherwise).*

Todd–Coxeter enumeration

Let's look at the example $\langle a, b | a^2 = b^2 = (ab)^2 \rangle$ above. We can write the relations as $aab^{-1}b^{-1} = 1$ and $abab^{-1} = 1$.

Attached to G and the generating set $\{a, b\}$ is a graph X, which is a directed graph (more precisely multigraph), the *Cayley graph*, defined as follows:

- The vertices are the elements of G.
- The arrows are labelled with a or b; there is an a-labelled arrow from g to h if $ga = h$, and similarly for b.

The relations imply that if we start at any vertex of X and "walk along the relation" – e.g. going from $g \to ga \to gab \to gaba \to gabab^{-1}$ – we have to get back to g again. The point of Todd–Coxeter enumeration is to gradually build up the Cayley graph of G, starting with the identity and gradually adding vertices and edges, while making the following promise: at each stage of the algorithm, if we walk from any vertex along a relation path, we will *either* fall off the edge of the graph, or we will get back to the starting vertex.

Let's see how this works for my example group. We'll give numbers to the elements of G in the order we discover them.

- At the start we know there is a vertex for the identity (number 1). We know that each of our two relations gives a path from 1 going back to 1 (but we don't know what happens in between). So we write the following in our notes, leaving gaps for the unknown parts:

	a	a	b^{-1}	b^{-1}
1				1

	a	b	a	b^{-1}
1				1

- The first "gap" is because we don't know where a sends element 1. So let's call this element 2. We now need a line in each table for the path starting at 2; and we make an extra table to record the fact that a sends 1 to 2:

	a	a	b^{-1}	b^{-1}
1	2			1
2				2

	a	b	a	b^{-1}
1	2			1
2				2

	a	b
1	2	

- The next thing we need is $2 \cdot a$, let this be 3:

	a	a	b^{-1}	b^{-1}
1	2	3		1
2	3			2
3				3

	a	b	a	b^{-1}
1	2			1
2	3			2
3				3

	a	b
1	2	
2	3	
3		

- Element 4 is going to be $3 \cdot b^{-1}$. Since the first relation has to work, we get for free that $4 \cdot b^{-1} = 1$, i.e. $1 \cdot b = 4$; so we can fill in a couple more entries in our tables using this:

	a	a	b^{-1}	b^{-1}
1	2	3	4	1
2	3			2
3				3
4			3	4

	a	b	a	b^{-1}
1	2		4	1
2	3			2
3				3
4		2	3	4

	a	b
1	2	4
2	3	
3		
4		3

- Fast-forward a few more steps; after 7 cycles we have a table

	a	a	b^{-1}	b^{-1}
1	2	3	4	1
2	3	5	6	2
3	5	7		3
4			3	4
5	7			5
6	4			6
7				7

	a	b	a	b^{-1}
1	2	6	4	1
2	3		6	2
3	5			3
4		2	3	4
5	7			5
6	4	3	5	6
7				7

	a	b
1	2	4
2	3	6
3	5	
4		3
5	7	
6	4	5
7		

To fill in the next blank in the first table, we need an element 8 with $8b = 7$ and $3b = 8$. Then we can put an 8 in the blank in the second table too; and we deduce $8a = 6$. So the new 8th row in the first table fills itself in: $8a = 6$, $6a = 4$, $4b^{-1} = 1$. So this row has to start as 8 6 4 1. But we also know it ends 7 8. So we have a *collapse*: elements 1 and 7 are the same!

So we can strike out all the 7's and replace them with 1's, and again apply all the relations we know. Then we get this table:

	a	a	b^{-1}	b^{-1}
1	2	3	4	1
2	3	5	6	2
3	5	1	8	3
4		8	3	4
5	1	2		5
6	4		5	6
-	-	-	-	-
8	6	4	1	8

	a	b	a	b^{-1}
1	2	6	4	1
2	3	8	6	2
3	5		8	3
4		2	3	4
5	1	4		5
6	4	3	5	6
-	-	-	-	-
8	6	5	1	8

	a	b
1	2	4
2	3	6
3	5	8
4		3
5	1	
6	4	5
-	-	-
8	6	1

- We're nearly there, but there are a few loose ends, we don't know $4a$ for instance. We'll introduce an element $4a = 9$ and see what happens:

	a	a	b^{-1}	b^{-1}
1	2	3	4	1
2	3	5	6	2
3	5	1	8	3
4	9	8	3	4
5	1	2	9	5
6	4	9	5	6
-	-	-	-	-
8	6	4	1	8
9	8	6	2	9

	a	b	a	b^{-1}
1	2	6	4	1
2	3	8	6	2
3	5	9	8	3
4	9	2	3	4
5	1	4	9	5
6	4	3	5	6
-	-	-	-	-
8	6	5	1	8
9	8	1	2	9

	a	b
1	2	4
2	3	6
3	5	8
4	9	3
5	1	9
6	4	5
-	-	-
8	6	1
9	8	2

We have no loose ends left: the relations imply that the subset of G consisting of these 8 elements is closed under multiplication by a or b, so in fact it's all of G. The conclusion is that G has order 8, and we can read off the group operation from our third table.

Cosets of subgroups

A more general version of Todd–Coxeter enumeration comes up when we have a presentation for G, and a list of elements (expressed as words in the generators) that generate a subgroup H. In this version, the assumption is that $[G : H]$ is finite (we don't suppose G iself is finite), and we want to compute a set of representatives for this quotient, together with the right-multiplication action of G on the set $H \backslash G$. This corresponds to gradually building up the *Schreier graph* of G and H, which is a generalisation of the Cayley graph with vertices indexed by cosets $H \backslash G$.

I won't go through the details, but the idea is very similar to the examples above (which is just the special case when $H = \{\mathrm{id}_G\}$); in this version, the numbers in our table represent cosets Hg_i (rather than just elements g_i). The only major difference is that we have an extra table for each generator of H, corresponding to the fact that H fixes the coset $\{1\}$; but H won't necessarily fix all the other cosets, since H might not be normal. So these "subgroup tables" always have a single row, starting and ending with 1 – we don't add extra rows to them.

Remark. A useful consequence of this algorithm is that it gives a **membership test** for the subgroup H: if we are given an element $g \in G$, expressed as a word in the generators, then Todd–Coxeter allows us to compute the permutation of $H \backslash G$ given by the action of g. This permutation fixes the coset of id_G iff $g \in G$.

11.2 Black box subgroups

We'll now consider the following variant of the Todd–Coxeter problem. As before, we have a finitely-generated group G, and a finite-index subgroup H. However, rather than a generating set for H, we have some black-box algorithm which takes an element of G, expressed as a word in the generators, and tells us if it's in H. Can we compute the index $[G : H]$ and a generating set of H?

Example. A key example is the *modular group* $G = \mathrm{SL}_2(\mathbb{Z})$. This has a presentation $G \cong \langle a, b \mid a^3 = b^2, a^6 = 1 \rangle$ where

$$a = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Suppose H is "all matrices in $\mathrm{SL}_2(\mathbb{Z})$ that are congruent to 1 mod 7". It's obvious this has finite index (it's the kernel of a homomorphism to a finite group), and if we have a word in a and b, we can compute the corresponding matrix and check if it's 1 mod 7.

We'll build a list X of representatives of $H \backslash G$ as follows. Suppose G has two generators a, b (as in my example). We start with $X = \{\mathrm{id}_G\}$. At each step, we compute, for each $x \in X$, the element xa. We want to know if the coset Hxa is one we already know, or not; so for each $y \in X$ we compute xay^{-1} and check if it's in H. If none of them are in H, then xa is a new coset, at we put it into X. If we don't discover any new cosets this way, we try cosets of the form xb instead. Since H has finite index, the process must stop: we get an X such that $\{Hx : x \in X\}$ is a subset of $H \backslash G$ stable under a and b, so it must be everything[1].

[1] Exercise: Why is this subset necessarily stable under a^{-1} and b^{-1} as well?

Remark. Note that in this algorithm we don't have the "collapsing" phenomenon: at each step the list of known cosets can only get bigger. So one can write down *a priori* bounds for the number of steps needed in terms of $[G : H]$.

The algorithm terminates with X a list of representatives of $H\backslash G$. However, it also gives us a little more. Say we number the elements of X as $x_1, \dots, x_n$, with $x_1 = 1$. As part of the above algorithm, for each $x_i \in X$, we've computed which H-coset contains $x_i a$; so we can write

$$x_i a = h_i x_{\alpha(i)}, \qquad h_i \in H,\ \alpha(i) \in \{1, \dots, n\}.$$

So we get a permutation α of $\{1, \dots, n\}$, and a bunch of elements h_i, recording how a acts on $H\backslash G$. Similarly, we get a permutation β and some elements $h'_i \in H$ recording how b acts.

Proposition. *The set of elements $\Sigma_H = \{h_1, \dots, h_n, h'_1, \dots, h'_n\}$ generate H.*

Proof. Let Σ_G be the given generating set of G. In the course of the above computation, we've computed a set of representatives X and, for each $u \in \Sigma_G$ and $x \in X$, a relation of the form $xu = vx'$ with $v \in \Sigma_H$. Simple manipulations also allow us to write $xu^{-1} = v^{-1}x''$ for some $v \in \Sigma_H$ and $x'' \in X$.

Let $h \in H$. By assumption we can write h as a word in the elements of Σ_G, say $h = u_1 u_2 \dots u_r$ where each u_i is in $\Sigma_G^{\pm 1}$. We have $x_1 = 1$, so $h = x_1 h = x_1 u_1 \dots u_r$; we can now use the above relations to shift the x step-by-step to the right, until we end up with a formula $h = v_1 \dots v_r x_j$ for some j, with $v_i \in \Sigma_H^{\pm 1}$. But $h \in H$, so x_j must be 1 and so we've written h using the elements of Σ_H. □

Remark. As well as a generating set for H, this also gives us an algorithm to solve the *word problem* for H, i.e. write any element of H in terms of the generators (assuming we already know how to do this for G). With a little extra work, we can also find a complete set of relations between the generators Σ_H, showing that finite-index subgroups of finitely-presented groups are finitely-presented.

11.3 Implementations

The tool we're most used to, Sage, is unfortunately not very good at computing with presentations of groups. For these problems, you are better off with Magma, or GAP (a specialist group theory package). Fortunately GAP comes bundled free with Sage: you can start it with `sage -gap` instead of `sage` at a terminal prompt.

12 Group theory II: Representations of groups

12.1 The setting

In this section G will be a *finite* group.

Definition. *A* representation *of G is a homomorphism $\rho : G \to \mathrm{GL}_n(\mathbb{C})$, for some n. Two representations are* isomorphic *if they differ by a change of basis.*

We say ρ is irreducible *if there is no nonzero subspace of $\mathbb{C}^n$ preserved by all the matrices $\rho(g)$ for $g \in G$.*

The theory tells us that any representation is isomorphic to a direct sum of irreducible representations (uniquely up to the order of the factors). Morever, the set of irreducible representations of G is finite.

A lot of information about ρ can be encoded in its *character*, which is the function $\chi_\rho : G \to \mathbb{C}$ defined by $\chi_\rho(g) = \operatorname{Tr} \rho(g)$. We have $\chi_\rho(h^{-1}gh) = \chi_\rho(g)$ for any $g, h \in G$, so χ_ρ is really a function on the set of *conjugacy classes* of G (which is usually much smaller than G itself, e.g. Conway's Monster group has order about 10^{54} but only a hundred or so conjugacy classes). It always takes values in the ring of integers of $\mathbb{Q}(\zeta_N)$, where N is the exponent of G (the largest order of an element); often it actually lies in a much smaller subring.

The *character table* of G is a grid with columns labelled by the conjugacy classes $[g_j]$ of G, and rows labelled by the irreducible representations ρ_i; the (i, j)-entry is $\chi_{\rho_i}(g_j)$. It turns out that it's a square grid, i.e. the number of representations is the number of characters (and in fact it's a matrix of full rank). This table encodes a huge amount of information about the structure of G.

Remark. The finite *simple* groups have been classified: there are a number of infinite families, and a list of "sporadic" groups not fitting into any family, of which Conway's Monster is the largest. There is an amazing book, the *Atlas of Finite Groups* (written by six of the leaders of the classification project), which contains – among other data – character tables for all the sporadic finite simple groups.

12.2 Burnside's algorithm

Our last topic will be an algorithm which computes the character table of a group G. We'll assume G is small enough that listing all its elements is practical (so this probably won't work for the Monster!).

Definition. *For R, S, T conjugacy classes in G, define*

$$c_{R,S,T} = \#\{(r,s) \in R \times S : rs = t\},$$

for t any element of T (it doesn't matter which we pick).

If we know the multiplication table of G, then we can compute the integers $c_{R,S,T}$.

Lemma. *If $\chi = \chi_\rho$ is the character of a d-dimensional irreducible representation, then for any two conjugacy classes R, S, we have*

$$\left(\frac{|R|\chi(R)}{d}\right) \cdot \left(\frac{|S|\chi(S)}{d}\right) = \sum_T c_{R,S,T} \left(\frac{|T|\chi(T)}{d}\right) .$$

Equivalently, if we number the conjugacy classes as $R_1, \ldots, R_r$, then for each $k = 1, \ldots, r$, the matrix M_k with (i,j)-th entry $c_{R_k R_i R_j}$ has the column vector v_ρ with j-th entry $\frac{|R_j|\chi_\rho(R_j)}{d}$ as an eigenvector. Note that if we number the conjugacy classes starting with $R_1 = \{\mathrm{id}_G\}$, then v_ρ has first entry 1.

Conversely, one checks that any vector that is a simultaneous eigenvector of all the M_k's must be a scalar multiple of v_ρ for some ρ. We can find such eigenvectors by diagonalising the M_k's, and this gives us enough information to reconstruct v_ρ (the scaling is fixed since the first entry of v_ρ is 1). The only thing left in order to reconstruct χ_ρ is to compute d; but this can be done using the "character orthogonality relations", which imply that

$$\sum_k \frac{|(v_\rho)_k|^2}{|R_k|} = \frac{|G|}{(\dim \rho)^2}.$$

Remark.

- Note that we don't need all the M_k's for this; we just need to take enough values of k to split up the space into 1-dimensional eigenspaces.
- Note that this gives us χ_ρ but it doesn't actually give us ρ as a map $G \to \mathrm{GL}_n$. This is a feature, not a bug: there are ways of getting your hands on explicit matrices if you really need them, but usually it's more informative to work with the characters of the representations rather than messing about with matrices.