# Finding orders with prescribed index in number fields

Alex J. Best
Supervised by Dr. Lassina Dembélé

2014

## Abstract

We develop algorithms for performing computations within algebraic number theory. Methods are developed to obtain all orders in a given number field with a specified index. We also apply these techniques to problems relating to elliptic curves over the rational numbers.

**Keywords.** Number theory, algebraic number theory, elliptic curves, number fields, algorithms.

# Contents

# 1 Introduction

This report deals with the algorithmic solution of a problem arising from algebraic number theory. Other interesting situations to which this algorithm can be applied are also discussed.

We first go through the background material needed to motivate, define and describe the solution of our problems in chapter 2. Then in section 3.1 we move on to the problems themselves and discuss the interest in studying them. After this in the rest of chapter 3 we detail the techniques used to solve the problems considered, starting with some special cases before moving onto more general results. Finally in section 4.1 we move on to some interesting applications of these methods to the study of elliptic curves.

# 2 Background material

In this section we fix several definitions and important results from algebraic number theory and commutative algebra. We assume only fairly basic knowledge of abstract algebra, such as the notions of groups, rings and fields.

The results given here are well known and are used throughout the rest of the report.

## 2.1 Commutative algebra

We introduce several notions and results that will be useful to us throughout the report, proofs for those results not proved here can be found in many textbooks on commutative algebra, such as [AM94] or [MR89].

All rings here are commutative with an identity element.

**Definition 1** (Module)**.** Given a ring $R$ we define an *$R$-module* $M$ to be an abelian group under addition, with a scalar multiplication map $\cdot : R \times M \to M$ satisfying

$$\begin{aligned} 1 \cdot m &= m \ \forall m \in M \\ r_1 \cdot (r_2 \cdot m) &= (r_1 r_2) \cdot m \ \forall r_1, r_2 \in R,\ m \in M \\ r \cdot (m_1 + m_2) &= r \cdot m_1 + r \cdot m_2 \ \forall r \in R,\ m_1, m_2 \in M \\ (r_1 + r_2) \cdot m &= r_1 \cdot m + r_2 \cdot m \ \forall r_1, r_2 \in R,\ m_1 \in M \end{aligned}$$

We often refer to elements of the ring $R$ as *scalars*.

From now on we will omit the notation $\cdot$ for scalar multiplication as it will be clear from context when the multiplication is taking place in the ring $R$ or on a module $M$.

The most common modules we will use here will be $\mathbb{Z}$-modules such as:

**Example 1.**

In fact every abelian group can be given a $\mathbb{Z}$-module structure so of course they are ubiquitous in all commutative algebra, nevertheless it is useful to think of the modules used here in terms of $\mathbb{Z}$ rather than as abstract abelian groups.

**Definition 2.** A *submodule* $N$ of a module $M$ is a module whose elements all belong to $M$ and whose operations are the same as those from $M$ on the elements on $N$.
Given two $R$-modules $M_1$ and $M_2$ we define their direct sum (denoted $\oplus$) to be the module with element set $M_1 \times M_2$ and operations given by the operations of $M_1$ and $M_2$ performed elementwise.

We now define a property of modules that makes them easier to work with and importantly easier to do computations with.

**Definition 3** (Finitely generated)**.** An $R$-module $M$ is *finitely* generated if there is a *finite* set $B \subset M$ such that any $m \in M$ can be written as

$$m = \sum_{b \in B} \alpha_b b$$

for some coefficients $\alpha_b \in R$.

**Definition 4** (Torsion submodule)**.** The *torsion submodule* $M_{\text{tors}}$ of an $R$-module $M$ is the set

$$\{m \in M \mid \exists r \in R \setminus 0 \text{ s.t. } rm = 0\}.$$

This is the set of elements that can be killed by a non-zero scalar. As implied by the name this is always submodule of $M$.

A module $M$ is called a torsion module if $M_{\text{tors}} = M$, and torsion-free if $M_{\text{tors}} = 0$.

**Definition 5** (Rank of a $\mathbb{Z}$-module)**.** Given a $\mathbb{Z}$-module $M$ we can always write $M = M_{\text{tors}} \oplus \mathbb{Z}^r$ for some unique $r \in \mathbb{Z}_{\geq 0}$. This $r$ is called the *rank* of the $\mathbb{Z}$-module.

**Example 2.**

$$M = \mathbb{Z}^2 = \{(a, b) \mid a, b \in \mathbb{Z}\}$$

is a torsion free $\mathbb{Z}$ module with submodule

$$N = 2\mathbb{Z} \times \mathbb{Z} = \{(2c, d) \mid c, d \in \mathbb{Z}\}.$$

The quotient module

$$M/N \cong (\mathbb{Z}/2\mathbb{Z}) \times \mathbb{Z}$$

has torsion submodule equal to $\mathbb{Z}/2\mathbb{Z}$ and is of rank 1.

## 2.2 Algebraic number theory

Algebraic number began with the study of *algebraic numbers* but has since expanded to encompass a huge amount of mathematics involving the use of algebraic techniques to tackle number theoretic problems. As above, more details about anything not proved here can be found in any of the many texts on algebraic number theory, for example [NS10], [Lan94].

**Definition 6** (Number field)**.** A *number field* $K$ is a field that is also a finite dimensional $\mathbb{Q}$-vector space.

**Example 3.** We write $\mathbb{Q}(\sqrt{3})$ for the smallest field containing both $\mathbb{Q}$ and $\sqrt{3}$, it is clear that the set

$$\{a + b\sqrt{3} \colon a, b \in \mathbb{Q}\}$$

must be contained in such a field. But we can also see that this set is closed under addition, subtraction, multiplication and non-zero division, and hence this set is the field $\mathbb{Q}(\sqrt{3})$.

Here we see that $\mathbb{Q}(\sqrt{3})$ has dimension 2 as a vector space over $\mathbb{Q}$.

**Definition 7.** The dimension of a number field $K$ as a $\mathbb{Q}$ vector space is called the *degree* of $K$.

We say that number fields of degree 2, such as in example 3 above are *quadratic*. Similarly degree 3 number fields are called *cubic*.

The following few definitions are central to the whole problem.

We can see that a number field will often have a large number of subrings, which may be of interest to us, however not all subrings are as nice as we would like them to be. So we distinguish some subrings that have desirable properties and single them out for study.

**Definition 8** (Order)**.** An *order* of a number field $K$ is a subring of $K$ that is finitely generated as a $\mathbb{Z}$-module, and of rank equal to the degree of $K$ (this is the maximal rank).

**Definition 9** (Ring of integers)**.** The *ring of integers*, denoted $\mathbb{Z}_K$, of a number field $K$ is the unique maximal order.

The terminology for this ring comes from the fact that its behaviour is analogous to the way $\mathbb{Z}$ behaves inside $\mathbb{Q}$.

**Definition 10** (Index)**.** Given two $R$-modules $M \subset N$ the index of $M$ in $N$, denoted $[N \colon M]$ is the size of the quotient abelian group, ignoring the module structure.

We can use the notion of index in a variety of situations, such as considering the index of a module in another, the index of a module in a ring, etc.

**Example 4.**

$$[\mathbb{Z} \cdot 1 + \mathbb{Z} \cdot \sqrt{2} \colon \mathbb{Z} \cdot 1 + \mathbb{Z} \cdot 2\sqrt{2}] = 2.$$

## 2.3 Elliptic curves

We now introduce the background material relevant to elliptic curves, one of the intended applications of these methods. This is not required for chapter 3, but section 4.1 uses these ideas heavily.

We are now ready to state in precise terms the project aimed to solve and to detail the methods used in its solution.

# 3 The problem

## 3.1 Statement

The aim of the project was to find a general method to solve the following problem, and moreover to find efficient algorithms that can solve the problem on any given inputs.

**Problem 1.** Given an order $R$ of an absolute number field $K$ and an integer $I$ find the set

$$\{\mathcal{O} \subseteq R \mid \mathcal{O} \text{ is a suborder, } [R\colon \mathcal{O}] = I\}.$$

To find a suborder we really mean compute a $\mathbb{Z}$ basis for the order, as such a basis defines an order completely.

One very natural extension of the above problem is to consider relative extensions of number fields. More precisely we wish to study the following problem.

**Problem 2.** Given an extension of number fields $L|K$, a $\mathbb{Z}_K$-order $R$ of $\mathbb{Z}_L$ and an integer $I$ find the set

$$\{\mathcal{O} \subseteq R \mid \mathcal{O} \text{ is a } \mathbb{Z}_K\text{-suborder, } [R\colon \mathcal{O}] = I\}.$$

We now detail some methods to solve this problem in increasing generality.

## 3.2 Quadratic number fields

Quadratic number fields are the simplest non-trivial number fields and they have a large amount of structure which can often make them easier to work with than more general number fields.

It is well known [Lan94] that the ring of integers of a quadratic field $K = \mathbb{Q}(\sqrt{d})$ for $d$ non-square always takes the form

$$\mathbb{Z}_K = \mathbb{Z} + \mathbb{Z}\alpha, \text{ with } \alpha = \begin{cases} \sqrt{d} & \text{if } d \equiv 2, 3 \pmod 4, \\ \frac{1+\sqrt{d}}{2} & \text{if } d \equiv 1 \pmod 4. \end{cases}$$

Indeed there is so little room for manoeuvre here that the following result on the structure of an order holds in this case.

**Proposition 1.** *Every order $\mathcal{O}$ of a quadratic number field can be expressed as*

$$\mathcal{O} = \mathbb{Z} + \mathbb{Z}f\alpha$$

*for some $f \in \mathbb{Z}$, $\alpha$ as above.*

For a proof see [Cox13, pp. 133–134].

**Definition 11.** The $f$ appearing in the above proposition is called the *conductor* of the order $\mathcal{O}$. Later we shall abuse this definition slightly by redefining the conductor to generalise this concept.

Now it is clear that

$$[\mathbb{Z} + \mathbb{Z}\alpha \colon \mathbb{Z} + \mathbb{Z}f\alpha] = |\mathbb{Z}/f\mathbb{Z}| = |f|.$$

So we have an incredibly simple solution for quadratic number fields, there is only one order of a given index.

## 3.3 Monogenic orders

## 3.4 Cocyclic orders

**Theorem 1.**

## 3.5 Absolute number fields

We originally hoped that the correspondence between suborders and their conductors that exists in the quadratic case (theorem 1) could be generalised to higher degree number fields. However the direct generalisations of this result fail to hold even in degree 3 number fields. We now give examples of some results that would be good for our purposes if true and explicit counter examples for each of them.

## 3.6 Relative number fields

## 3.7 Practical speed-ups

Though the methods described above are the best we have been able to obtain so far from a theoretical perspective, when it comes down to computing examples in the real world there are a number of ideas and techniques we can use to reduce the time taken.

### 3.7.1 Parallelisation

Several stages of the algorithm for general orders can be naturally parallelised reducing the total time taken linearly with the number of threads used. For example once the set of potential conductors has been computed the orders to which they correspond can all be computed in parallel.

# 4 Applications

Through the main problem itself is an interesting one which is worth studying in its own right we were also motivated to look at it by the potential applications to other questions within the same areas of mathematics. One of the most prominent areas in which a solution to the problem can be used is to answer questions about elliptic curves. How a solution to the problem considered above can be applied in this case is detailed below, along with results obtain from the application of our methods there. Also ???

## 4.1 Elliptic curves

Given an elliptic curve over the rationals in long Weierstrass form such as

$$E_l\colon y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_1x + a_0 \text{ with } a_i \in \mathbb{Q}$$

we can find an isomorphic curve $E\colon y^2 = x^3+$.

Given such a curve we can define a number field to help us study the torsion points of $E$.

**Definition 12.** For a prime $p$ the $p$-division field of $E$, denoted $\mathbb{Q}(E[p])$ is obtained by taking all points $(x, y) \in E[p] \setminus 0 \subset \mathbb{Q}^2$ and adjoining each of their coordinates $x$ and $y$ to $\mathbb{Q}$.

As the torsion subgroup $E_{\text{tors}}$ is always finite there are at most finitely many elements that need adjoining to $\mathbb{Q}$, so this field is fairly nice.

Assuming that the curve $E$ has no rational 2-torsion points we must have no rational solution to $f = x^3+??$ and hence the polynomial is irreducible over $\mathbb{Q}$. We can then see that we have

$$\mathbb{Q}(E[2]) \cong \mathbb{Q}[x]/(f).$$

Letting $\alpha$ be the image of $x$ in the above quotient, we can see that

$$\mathbb{Z}[\alpha] \subset \mathbb{Q}(\alpha)$$

is an order of $\mathbb{Q}(\alpha)$.

# 5 Conclusion

## 5.1 Further work

## 5.2 Acknowledgements

First and foremost I would like to thank Lassina Dembélé for his excellent guidance while I undertook the project. I am also grateful to John Cremona for allowing me access to one of the Warwick number theory group's servers to run computations on.

# 6 Appendix: Code

Much of what was done has been implemented in both the Sage and Magma and so we provide annotated source code listings for the algorithms in both languages below.

## 6.1 Sage

```
def conductor(order): # Seems correct now
    """
    Return the conductor of the order.
    """

    K = order.fraction_field()
    R = Integers()
    ZK = K.maximal_order()
    omega = ZK.basis()
    n = order.rank()
    M = matrix(R.fraction_field(), nrows = n*n, ncols = n)

    d = 1
    for i in range(n):
        for j in range(n):
            coords = order.coordinates(ZK.gen(i)*ZK.gen(j))
            for k in range(n):
                M[j*n + k, i] = coords[k]
                d = lcm(d, coords[k].denominator())

    H = (d * M).change_ring(R).hermite_form(include_zero_rows = False)

    return K.ideal(list(vector(omega) * d * H.inverse()))

def orders_of_index(O,I):
    """
    Returns a list of orders with the given index.
    """
    K = O.fraction_field()
    ZK = K.maximal_order()
    R = Integers() # This may need to be different when relative orders are
        looked for.

    # We find all ideals of norm dividing I^2 and divisible by I, which are
        all possibly conductors of our order.
    possible_conductors = []
    for d in divisors(I):
        possible_conductors += ideals_of_norm(K, I*d)
```

```
    print possible_conductors

    orders = []
    for f in possible_conductors:
        #print f
        cur_orders = orders_with_conductor_and_index(f, I)
        if cur_orders:
            orders += cur_orders
        print str(len(cur_orders)) + " order(s) found with right index."
    return orders

def ideals_of_norm(K,N): # Correct?, could use less memory?
    # We use the factorisation of prime ideals to find all ideals with
        given norm N
    primes = dict() # primes[p][i] will contain all prime ideals with norm
        p^i.
    ideals = [K.ideal(1)]
    for (p,e) in N.factor():
        #print str(p)+"^"+str(e)
        primes[p] = dict()
        for i in range(1,e+1):
            primes[p][i] = []

        for (P,E) in K.factor(p): # for P in K.prime_factors(p) ?
            v = valuation(P.norm(), p)
            # replacing the condition below with v <= e would probably be
                faster, if it turns out we cannot rule out any k <= e as
                exponents.
            if v in primes[p]: # Otherwise the ideal is too small to be of
                use to us.
                primes[p][v].append(P)

        possible_factorisations = []
        for partition in Partitions(e, parts_in = primes[p].keys()):
            #print partition
            current_factorisation = [K.ideal(1)]
            for i in partition:
                new_factorisation = []
                for P in primes[p][i]:
                    for f in current_factorisation:
                        new_factorisation.append(f*P)
                current_factorisation = new_factorisation
            possible_factorisations += new_factorisation

        current_ideals = []
        for P in possible_factorisations:
            for f in ideals:
                current_ideals.append(f*P)
        ideals = current_ideals
    return ideals

def orders_with_conductor_and_index(f,I):
    K = f.number_field()
```

```
    Zk = K.maximal_order()
    naive_O = K.order(f.basis()) # Use ring_generators here?!
    if conductor(naive_O) == f:
        if naive_O.index_in(Zk) == I:
            return [naive_O]
        else: # Still not clever enough!
            orders = []
            quo = Zk.free_module().quotient(naive_O.free_module())
            r = quo.cardinality() / I
            for a in quo:
                if a.additive_order() == r:
                    O = K.order(naive_O.gens() + [Zk(a.lift())])
                    if O.index_in(Zk) == I:
                        if not O in orders:
                            orders.append(O)
            return orders
    else:
        pass
        #raise Exception("AAAH")
    return []

def cocyclic_orders_of_index(O,I):
    """
    Returns a list of orders with the given index.
    """
    K = O.fraction_field()
    ZK = K.maximal_order()
    R = Integers() # This may need to be different when relative orders are
         looked for.

    possible_conductors = ideals_of_norm(K, I*I)

    #print possible_conductors

    orders = []
    for f in possible_conductors:
        q = ZK.free_module().quotient(f.free_module())
        if q.ngens() == 2:
            if q.gens()[0].order() == I: # ????
                orders.append(cocyclic_order_with_conductor(f))
        #print f
        #print f.basis()
    return orders

def cocyclic_order_with_conductor(f):
    K = f.number_field()
    Zk = K.maximal_order()
    O = K.order(f.basis()) # Use ring_generators here?!
    if conductor(O) == f:
        return O
    raise Exception("Order was not cocyclic.")
    return []
```

## 6.2 Magma

```
function overOrder(I,K) // Returns the smallest order containing I
    return Order(Basis(I,K));
    //return Order(TwoElement(I));
end function;

function idealOfO(O,I)
    return ideal<O|[O!b : b in Basis(I,FieldOfFractions(O))]>;
end function;

function orderOfIndex(K, d)
    "Finding orders:";
    primeIdeals := AssociativeArray();
    Zk := Integers(K);

    for pe in Factorisation(d^2) do
        p := pe[1];
        e := pe[2];
        "Prime:", p;
        primeIdeals[p] := AssociativeArray();

        for i in [1..e] do
            primeIdeals[p][i] := [];
        end for;

        for PE in Factorisation(ideal<Zk|p>) do
            P := PE[1];
            E := PE[2];
            v := Valuation(Norm(P),p);
            if v in Keys(primeIdeals[p]) then // Otherwise the exponent is
                too high anyway
                Append(~primeIdeals[p][Valuation(Norm(P),p)],P); // Add P
                    to the set of prime ideals of norm p^e
            end if;
        end for;
        for x in Keys(primeIdeals[p]) do x, primeIdeals[p][x]; end for;
    end for;

    "Partitions:";

    possConds := [ ideal<Zk|1> ];
    for pe in Factorisation(d^2) do
        p := pe[1];
        e := pe[2];
        p,"^",e;
        possFacts := [ ];

        for part in RestrictedPartitions(e, Keys(primeIdeals[p])) do
            part;
            curFact := [ideal<Zk|1>];
            for i in part do
                newFact := [ ];
```

```
                for P in primeIdeals[p][i] do
                    for f in curFact do
                        Append(~newFact, f*P);
                    end for;
                end for;
                curFact := newFact;
            end for;
            possFacts cat:= curFact;
        end for;
        curConds := [];
        for P in possFacts do
            for f in possConds do
                Append(~curConds, f*P);
            end for;
        end for;
        possConds := curConds;
    end for;

    out := [];

    "Trying conductors:";
    for f in possConds do
        f,Norm(f);
        Of := overOrder(f,K);
        if Conductor(Of) eq f then
            if Index(Zk,Of) eq d then // Sanity check
                Append(~out, Of);
            else;
                //error Error();
            end if;
        end if;
    end for;

    return out;
end function;
```

# Bibliography


[AM94] M. Atiyah and I. G. MacDonald. *Introduction to Commutative Algebra*. Addison-Wesley series in mathematics. Westview Press, 1994.

[Bra09] J. Brakenhoff. *Counting problems for number rings*. PhD thesis, Universiteit Leiden, 2009.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer, 1993.

[Coh00] H. Cohen. *Advanced Topics in Computational Number Theory*. Graduate Texts in Mathematics. Springer New York, 2000.

[Cox13] D. A. Cox. *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory, and Complex Multiplication*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2013.

[Lan94] S. Lang. *Algebraic Number Theory*. Applied Mathematical Sciences. Springer, 1994.

[MR89] H. Matsumura and M. Reid. *Commutative Ring Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1989.

[NS10] J. Neukirch and N. Schappacher. *Algebraic Number Theory*. Grundlehren der mathematischen Wissenschaften. Springer, 2010.

[PZ89] M. Pohst and H. Zassenhaus. *Algorithmic Algebraic Number Theory*. Cambridge University Press, 1989.