Computing Cubic Fields in Quasi-Linear Time

K. Belabas

Département de mathématiques (A2X)
Université Bordeaux I
351, cours de la Libération, 33405 Talence (France)
belabas@math.u-bordeaux.fr

Cubic fields (over the rationals) are the simplest non-Galois number fields and thus should be the ideal testing ground for most general "density" conjectures, such as the Cohen-Martinet heuristics. We present an efficient algorithm to generate them, up to a given discriminant bound, which we hope will prove a useful tool in their computational exploration.

It all originates from the seminal paper [4] by Davenport and Heilbronn and some reduction theory as was already known to Hermite. When no explicit reference has been given, we refer the curious reader not wishing to consider the proofs as (easy) exercises to [1].

The rationale is as follows: to a given cubic field, we associate first a class of binary cubic forms, which shares the same discriminant, and then a canonical representative in the class. The essential point is that we have an explicit description of the image of this mapping, the set of companion forms, which behaves nicely from the algorithmic point of view.

1. The Theory

We consider Φ the set of integral, irreducible, primitive, binary cubic forms. One classically associates to the form $F(x,y) = ax^3 + bx^2y + cxy^2 + dy^3$, or $F = (a,b,c,d)$ in short, its discriminant

$$\operatorname{disc}(F) = b^2c^2 - 27a^2d^2 + 18abcd - 4ac^3 - 4b^3d \ .$$

A linear change of variables defines a natural, discriminant-preserving, action of $\mathrm{GL}_2(\mathbb{Z})$ on Φ, and we call $\overline{\Phi}$ the quotient set. Define the Hessian form of F,

$$H_F = -\frac{1}{4}\begin{vmatrix} \frac{\partial^2 F}{\partial x \partial x} & \frac{\partial^2 F}{\partial x \partial y} \\ \frac{\partial^2 F}{\partial x \partial y} & \frac{\partial^2 F}{\partial y \partial y} \end{vmatrix} = Px^2 + Qxy + Ry^2 \ ,$$

$$\text{with } P = b^2 - 3ac,\ Q = bc - 9ad,\ R = c^2 - 3bd.$$

This is a covariant quadratic form (*i.e.* $H_{F\circ M} = H_F \circ M$ for all M in $\mathrm{GL}_2(\mathbb{Z})$) whose discriminant is $-3\operatorname{disc}(F)$. We call $f_H(F)$ the Hessian content, that is the gcd of (P,Q,R).

For every prime p, we introduce, following Davenport and Heilbronn, some sets of classes of forms, V_p and U_p. V_p denotes the subset of $\overline{\Phi}$ whose elements satisfy

$$\begin{cases} \operatorname{disc}(F) \equiv 1 \mod 4 \text{ or } \operatorname{disc}(F) \equiv 8 \text{ or } 12 \mod 16 & \text{if } p = 2, \\ \operatorname{disc}(F) \not\equiv 0 \mod p^2 & \text{otherwise.} \end{cases}$$

If $p \neq 3$, we take

$$U_p = \left\{ F \in \overline{\Phi} : F \in V_p \text{ or } \left[p | f_H(F) \text{ and } p^3 \nmid \operatorname{disc}(F) \right] \right\} .$$

The set U_3 is a little more complicated. We decide whether the form $F = (a, b, c, d)$ belongs to U_3 according to the following algorithm:

$$\begin{array}{rll} \text{if} & F \in V_3, & \text{then } F \in U_3, \\ \text{else if} & 3 \nmid f_H, & \text{then } F \notin U_3, \\ \text{else if} & 3 | a, & F \in U_3 \Longleftrightarrow 9 \nmid a \text{ and } 3 \nmid d, \\ \text{else if} & 3 | d, & F \in U_3 \Longleftrightarrow 9 \nmid d, \\ \text{else if} & 3 | (a-d), & F \in U_3 \Longleftrightarrow a - b + c - d \equiv 0 \pmod 9, \\ \text{else if} & 3 | (a+d), & F \in U_3 \Longleftrightarrow a + b + c + d \equiv 0 \pmod 9. \end{array}$$

Davenport and Heilbronn defined U_p in a different (more elegant) way, but our presentation is best-suited for our algorithmic purpose. One can show that a primitive F does not belong to U_p if and only if $F(x, y)$ has a square factor $(\beta x - \alpha y)^2$ modulo p, with $F(\alpha, \beta) \equiv 0 \pmod{p^2}$. We set $U = \cap U_p$.

Theorem 1 (Davenport-Heilbronn)**.** *The classes of U are in one-to-one correspondence with the (isomorphism classes of) cubic extensions of $\mathbb{Q}$. This bijection associates to any representative $F = a(x - \tau_1 y)(x - \tau_2 y)(x - \tau_3 y)$ of a class the set of fields $\{\mathbb{Q}(\tau_1), \mathbb{Q}(\tau_2), \mathbb{Q}(\tau_3)\}$. Besides, the common discriminant of these fields is equal to* $\operatorname{disc}(F)$.

Remark.. *The reciprocal mapping is also explicit, given by the index form.*

Let $F = (a, b, c, d)$ be a cubic form of *positive* discriminant, and (P, Q, R) its Hessian. We call F reduced if $|Q| \leqslant P \leqslant R$ and

- $a > 0$, $b \geqslant 0$ and $d < 0$ whenever $b = 0$.
- If $Q = 0$, $d < 0$.
- If $P = Q$, $b < |3a - b|$.
- If $P = R$, $a \leqslant |d|$, and $b < |c|$ whenever $|d| = a$.

The corresponding notion for forms of *negative* discriminant is given by the following inequalities:

$$\begin{gathered} a > 0, \quad b \geqslant 0 \quad \text{and } d > 0 \text{ whenever } b = 0, \\ d^2 - a^2 + ac - db > 0, \\ -(a-b)^2 - ac < ad - bc < (a+b)^2 + ac. \end{gathered}$$

Theorem 2. *A reduced cubic form belonging to U is irreducible. Any irreducible cubic form is equivalent to a unique reduced one.*

Let X be a positive real number and $F = (a, b, c, d)$ a reduced form. Call $\theta(a, b, X)$ the unique positive real solution of the equation

$$-4\theta^3 + (3a + 2b)^2 \theta^2 + 27 a^2 X = 0 \ .$$

Lemma 3. *If $d(F)$ lies in $]0, X]$, we have:*

$$1 \leqslant a \leqslant \frac{2X^{1/4}}{3\sqrt{3}}, \quad 0 \leqslant b \leqslant \frac{3a}{2} + \sqrt{\sqrt{X} - \frac{27a^2}{4}},$$

$$\text{and} \quad \frac{b^2 - \theta(a,b,X)}{3a} \leqslant c \leqslant b - 3a \ .$$

Lemma 4. *If $d(F)$ belongs to $[-X, 0[$, we have*

$$1 \leqslant a \leqslant \left(\frac{16X}{27}\right)^{1/4}, \quad 0 \leqslant b \leqslant \frac{3a}{2} + \sqrt{\left(\frac{X}{3}\right)^{1/2} - \frac{3a^2}{4}},$$
$$\text{and} \quad 1 - b \leqslant c \leqslant \frac{b^2}{3a} + \left(\frac{X}{4a}\right)^1 /3.$$

By means of Theorems 1 and 2, we know we can associate a canonical companion form to each cubic field. Lemmas 3 and 4 tell us where to look, given a bound for the field discriminant. Incidentally, this process yields a canonical basis for the maximal order:

Lemma 5. *Let K be a cubic field, generated by a root θ of the cubic form $F = (a, b, c, d)$. Suppose the discriminants of F and K are equal. Then*

$$[1, a\theta, a\theta^2 + b\theta]$$

is a $\mathbb{Z}$-basis of the maximal order of K.

The lemma remains true, and no harder to prove, for fields of arbitrary degree n (consider binary n-forms with the correct discriminant instead). The cubic and quadratic cases are peculiar in that such forms always exist !

2. The Algorithms

2.1. **Common Routines.** We shall see that the algorithm implicit in the preceding section is linear in the discriminant bound X save for the time spent checking whether about X integers, of size bounded by X, are square-free. In order to reduce this factoring time, we pre-compute lists of "square-full" numbers (satisfying $p^2|\Delta$ for some $p > P$). Thus, the check for square-freeness reduces to a binary search in a sorted list followed by a few trial divisions: about $\pi(P)$, where $\pi(x)$ is the number of primes up to x. This reduces tremendously computational time.

Call M the maximum memory one is willing to spend for the list. This means we keep at most M 32-bit integers in RAM. This works nicely when $X < 2^{32}$. For larger X, we use 2^k lists and a primitive hashing technique, storing only the lowest order words, using the k highest order bits to choose the right list ($k = 10$ is more than enough). This leads to the following initialization routine:

Sub-Algorithm 1 (`init`)

(1)[Initialize primes]: Input X, the discriminant bound. Compute a table of primes up to $\sqrt{X}$, `p[]`, as well as their squares `pp[]`. Using a binary search, find the minimal prime p such that:

$$\frac{X}{p\log p}\cdot\left(1+\frac{1}{2\log p}\right)\leqslant 3M\ .$$

If $p\leqslant 53$, find the minimal prime p such that

$$\sum_{\substack{p\leqslant l\leqslant 53\\ l\text{ prime}}} l^{-2}\leqslant \frac{3M}{X}-\frac{1}{59\log 59}\cdot\left(1+\frac{1}{2\log 59}\right)\ .$$

If $p<5$, set $p=5$. Set `index` such that `p[index]`=p.

(2)[Initialize sieve]: Store in `list[]` all the integers less than X, prime to 6, and admitting a divisor `pp`$[i]$ for some $i\geqslant$ `index`, as explained above. Fill in boolean array `sqfull[]` up to $n=\sqrt{3X}$, such that `sqfull`$[n]$ is `true` if and only if n has a square factor prime to 6.

Lemma 2. *The choice of* `p[index]` *given in step* 1 *ensures that* `list[]` *will contain less than* M *integers.*

We use the following general purpose subalgorithm:

Sub-Algorithm 3 (`test`(f_H,a,b,c,d,Δ))

Input: $F=(a,b,c,d)$ a reduced cubic form belonging to U_2, f_H the content of its Hessian H, and $\Delta=|\operatorname{disc}(H)|$ (recall that $\Delta=-3\operatorname{disc}(F)$).
Output: F if it belongs to U, nothing otherwise.

(1)If (a,b,c,d) does not belong to U_3 or `sqfull`$[f_H]$ is `true`, return.

(2)Set $t=\Delta/f_H^2$, and remove all powers of 2 and 3 from the factorization of t. If $\gcd(t,f_H)>1$, return.

(3)Return if t is not square-free. This test should be done as follows: if n is small enough ($n\leqslant\sqrt{3X}$) return if `sqfull`$[n]$ is `true`. Else search the (sorted by construction) `list` for n, then trial divide n by `pp`$[i]$, $2\leqslant i<$ `index`, returning as soon as n is found or one `pp`$[i]$ divides n

(4)Output (a,b,c,d).

Lemma 4.

- *The integer* f_H *is less than* $\sqrt{3X}$*. Thus, if* $|\operatorname{disc}(F)|\leqslant X$*, the test for* `sqfull`$[f_H]$ *in step* 1 *is meaningful.*
- *If* $F=(a,b,c,d)$ *belongs to* U_2 *and* U_3*, then* $gcd(t,6^\infty)|72$ *in step* 2.

2.2. **Real Cubic Fields.** The following sub-algorithm checks whether the binary cubic form $ax^3+bx^2y+cxy^2+dy^3$ corresponds to a cubic field, using reduction theory specific to the real case:

Sub-Algorithm 5 (`is_real_field`(a,b,c,d,P,Q,R))

Input: a real cubic form $F = (a, b, c, d)$, and its reduced Hessian (P, Q, R).
Output: F, if it corresponds to a (real) cubic field.

(1)[Check special cases]:
- if $P = Q$: if $|b| \geqslant |3a - b|$, return.
- if $P = R$: if $a > |d|$, return. If $a = |d|$ and $|b| \geqslant |c|$, return.
- if $|Q| = R$: if $4|P$ return. Execute `test`$(P, a, b, c, d, 3P^2)$, then return.

(2)[$F \in U_2$?] Set $\Delta = 4PR - Q^2$. If $16|\Delta$ or [$\Delta \equiv 12 \pmod{16}$ and either P or R is odd], return.

(3)Set $f_H = \gcd(P, Q, R)$, then execute `test`$(f_H, a, b, c, d, \Delta)$.

Remark.. *Step 1 ensures that F is reduced. When $|Q| = R$, which implies $|Q| = P = R$, we are in the cyclic case,* i.e. *the companion field of F, if it exists, is cyclic (this is a necessary and sufficient condition). As we already know that f_H will be equal to P, we take a shortcut.*

It only remains to loop on the coefficients (a, b, c, d) of the cubic form, each time calling this procedure to check for cubic fields. Given a bound X for the discriminant, the constants (a, b, c) appearing next shall satisfy the inequalities in Lemma 3. Finally, given a, b, c and X, the integer d satisfies

$$|bc - 9ad| \leqslant b^2 - 3ac \leqslant c^2 - 3bd, \quad d < 0 \text{ if } b = 0 \tag{1}$$

for we want (a, b, c, d) to be reduced, and

$$(-27a^2) \cdot d^2 + 2(9ac - 2b^2)b \cdot d + c^2(b^2 - 4ac) \leqslant X \ , \tag{2}$$

because of the discriminant bound (given (1), expression (2) is non-negative).

Algorithm 6 (CRFCRF[1])

Input: a discriminant bound X.
Output: the forms associated to the real cubic fields whose discriminants are less than X.

(1)Execute `init`.

(2)[Special case $b = 0$] Execute three embedded loops on a, c, d, in this nesting order. Set the bounds using the preceding inequalities (which are much simpler in this case). Compute the Hessian (P, Q, R). which is reduced by construction, then execute `is_real_field`$(a, 0, c, d, P, Q, R)$.

(3)[General case] We now have four loops on a, b, c, d in this order, with the additional inequality $b > 0$. Compute the Hessian (P, Q, R), then execute `is_real_field`(a, b, c, d, P, Q, R).

Remark.. *Great care must be taken in setting the bounds for the various loops to avoid round-off errors. Also, many computations can be done at an early stage. For instance $P = b^2 - 3ac$ can – and should – be computed before d is known. This is tedious but essentially straightforward, so we chose not to hide*

[1]stands for Cubic Real Fields Counting Reduced Forms.

the simplicity of the algorithm behind scores of auxiliary variables and explicit complicated bounds.

2.3. **Complex Cubic Fields.** Looking for complex cubic fields whose (negative) discriminant is greater than $-X$, we arrange for (a, b, c) to satisfy the inequalities of Lemma 4. Now d must satisfy

$$d^2 - a^2 + ac - db > 0 \ , \tag{3}$$

$$-(a-b)^2 - ac < ad - bc < (a+b)^2 + ac \ , \tag{4}$$

$d > 0$ whenever $b = 0$ and, finally,

$$-X \leqslant (-27a^2) \cdot d^2 + 2(9ac - 2b^2)b \cdot d + c^2(b^2 - 4ac) < 0 \ . \tag{5}$$

This time, the reduction inequalities do not imply that the discriminant is negative.

Sub-Algorithm 7 (`is_complex_field`(a, b, c, d, P, Q, R))

(1)[$F \in U_2$?] Set $\Delta = Q^2 - 4PR$. If $16|\Delta$ or [$\Delta \equiv 4 \pmod{16}$ and either P or R is odd], return.
(2)Set $f_H = \gcd(|P|, |Q|, |R|)$, then execute `test`$(f_H, a, b, c, d, \Delta)$.

The shape of the algorithm is the same as in the real case. One must change the bounds as indicated above and use `is_complex_field` instead of its real counterpart. The new acronym is CCFCCF[2].

3. Complexity and General Remarks

3.1. Recall that Davenport ([2] and [3]) proved that the number of reduced forms whose discriminants are bounded by X is equivalent to

$$\frac{\pi^2}{72}X \quad \text{in the real case,} \quad \frac{\pi^2}{24}X \quad \text{in the complex case.} \tag{6}$$

It does happen that for given (a, b, c) satisfying our reduction bounds, there does not exist d such that the form (a, b, c, d) is both reduced and has a discriminant in the expected interval. One can show the number of these "empty loops" is a $O(X^{3/4})$. Thus the number of loops in our algorithms is equivalent to the number of reduced forms in the same discriminant range, given by (6).

Asymptoticaly, most of the time spent in a given loop will be taken by the `index` trial divisions used to locate small square factors when all else has failed. This number is bounded by a (small) constant times

$$\min\left(\frac{X}{3M\log^2 X}, \frac{X^{1/2}}{\log X}\right) \ ,$$

where M is the number of integers we can afford to store in the pre-computed `list`. As M is typically around 10^7, this remains small for the practical range of the method (less than 120 divisions for $X \leqslant 10^{11}$ in our implementation). Thus we claim the algorithm will run in quasi-linear time.

[2]Cubic Complex Fields Counting Companion Forms.

3.2. Davenport and Heilbronn [4] later proved that, given a bound X for their discriminants, the number of cubic fields is equivalent to

$$\frac{1}{12\zeta(3)}X \quad \text{in the real case,} \quad \frac{1}{4\zeta(3)}X \quad \text{in the complex case.}$$

This is about one half the corresponding values for reduced forms. Thus among our loops, only about one half will yield incorrect forms. Hence, there is very little waste.

3.3. One can sensibly compute the number of (isomorphism class of) cubic fields up to $X \approx 10^{11}$ in this way. As one can see from Table 4.1 below, the overhead computations in subroutine `init` take a negligible time, thus the algorithm can easily be distributed.

The intermediate results all fit in single precision (long) integers on 64-bit machines for reasonable X: say, less than 10^{12} in the real case, and 5.10^{10} in the complex case.

3.4. It is easy to compute fields whose discriminants lie in an interval $[X, X+Y]$, for large X (say 10^{14}), when Y is small enough (say 10^6). We incorporate the relevant discriminant inequality in the loops and, instead of using lists of square-full numbers, we factor the discriminant using a suitable probabilistic factorization method. The running time is then essentially the time needed to factor around Y numbers of size X. There are still at most $O(X^{3/4})$ "empty" loops though, and this can become dominant if X is too large.

3.5. If one compares with methods originating from Hunter's theorem, the gain is gigantic: no irreducibility check, no need to factor the discriminant, no search for automorphisms and thus no need to keep all the fields found so far in memory. As a matter of fact, sorting the field by increasing discriminant (which is utterly impossible if X is large) actually takes much more time than computing them.

4. Results

This algorithm has been implemented in ANSI C on a DEC alpha (a fast 64-bit machine). The following tables give an idea of computational time and memory usage in this case. First, we consider the `init` routine, which does not depend on the signature. Most of the time in there is spent building sieves. We call $P = \texttt{p[index]}$ the prime chosen to build the hashing lists. For instance, $P = 5$ means that no trial division actually takes place in `sqfree`. The "Square-full ints" column corresponds to the number of 32-bit integers stored in the hashing lists:

X	P	Square-full ints	Sieving time
10^4	5	290	0.001 s
10^5	5	2935	0.01 s
10^6	5	29370	0.1 s
10^7	5	293674	1.0 s
10^8	5	2936998	7.0 s
10^9	17	5474664	43 s
10^{10}	97	6409864	356 s (5 min 56 s)
10^{11}	661	6644929	3427 s (58 min 15 s)

TABLE 4.1: Overhead Computations

Next, we give the data corresponding to the computation of real and complex cubic fields. "a" denotes the maximal value for the first coefficient of the cubic form. These happen to be the ones given respectively by the bound in Lemma 3 and one less than the ones in Lemma 4 (with the exception $X = 10^4$ for the latter where we get the exact bound).

We get a roughly linear behavior as long as $P = 5$, which quickly "diverges" as P increases. Up to the same discriminant bound, time spent for the complex computations compared to the real ones should be in the same ratio as the number of fields found (slowly decreasing in the given examples, equal to 3 at infinity due to Davenport and Heilbronn's result). Hence they should be about three times slower (not exactly so, the initializing step being exactly the same). But the situation is a little worse, due to the extra square roots arising in the complex case: given (a, b, c, X), d must satisfy three quadratic inequalities instead of one (compare (2) with (5), and (1) with (3), (4)).

	X	# of fields	Elapsed time	a
	10^1	0	0.000 s	0
	10^2	2	0.000 s	1
	10^3	27	0.000 s	2
	10^4	382	0.005 s	3
	10^5	4,804	0.05 s	6
	10^6	54,600	0.5 s	12
	10^7	592,922	5.7 s	21
	10^8	6,248,290	64.6 s (1 min 05 s)	38
$P > 5$	10^9	64,659,361	774 s (12 min 54 s)	68
	10^{10}	661,448,081	18,641 s (5 h 11 min)	121
	10^{11}	6,715,824,025	797,373 s (9 days 5 h)	216

TABLE 4.2: Real cubic fields

	X	# of fields	Elapsed time	a
	10^1	0	0.000 s	0
	10^2	7	0.000 s	1
	10^3	127	0.004 s	3
	10^4	1520	0.04 s	7
	10^5	17,041	0.3 s	14
	10^6	182,417	2.2 s	26
	10^7	1,905,514	21.6 s	49
	10^8	19,609,185	224 s (3 min 44 s)	86
$P > 5$	10^9	199,884,780	2,575 s (42 min 55 s)	155
	10^{10}	2,024,660,098	58,803 s (16 h 20 min)	276
	10^{11}	20,422,230,540	2,427,276 s (28 days 2 h)	492

TABLE 4.3: Complex cubic fields

References

1. Belabas, K. A fast algorithm to compute cubic fields (to appear in *Math. Comp.*).
2. Davenport, H. On the class number of binary cubic forms (I), *J. Lond. Math. Soc.* **26** (1951), pp. 183–192. (erratum, *ibid* **27**, p. 512).
3. Davenport, H. On the class number of binary cubic forms (II), *J. Lond. Math. Soc.* **26** (1951), pp. 192–198.
4. Davenport, H. & Heilbronn, H. On the density of discriminants of cubic fields (II) *Proc.Roy.Soc.Lond.A* **322** (1971), 405-420.