

# 15 Number Theory

## 15.3 Positive Definite Binary Quadratic Forms (9 units)

*Background material is contained in the Part II course Number Theory. Part IB Groups, Rings and Modules and Part II Number Fields may be helpful, but are not necessary.*

### 1 Introduction

We start with a *binary quadratic form*  $f(x, y) = ax^2 + bxy + cy^2$  with  $a, b, c \in \mathbb{Z}$ , which we shall abbreviate as  $(a, b, c)$ . The *discriminant* of the form  $(a, b, c)$  is  $d = b^2 - 4ac$ . Note that  $d$  is always congruent to 0 or 1 modulo 4. We consider only *positive definite* forms, for which  $d$  is negative and  $a$  is positive.

Two forms  $f, g$  are *equivalent*, written  $f \sim g$ , if one can be transformed into the other by a *unimodular substitution*  $M$ , that is, if  $g(x, y) = fM(x, y) = f(sx + ty, ux + vy)$  where  $s, t, u, v \in \mathbb{Z}$  and  $sv - tu = 1$ , i.e.

$$M = \begin{pmatrix} s & t \\ u & v \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}).$$

Equivalent forms have the same discriminant, but the converse is not true in general. A form  $(a, b, c)$  is *primitive* if no integer greater than one divides all three of  $a, b$  and  $c$ .

### 2 Computing the class number

A form  $(a, b, c)$  is *reduced* if either  $-a < b \leq a < c$  or  $0 \leq b \leq a = c$ . There are only finitely many reduced forms of given discriminant. It is known that distinct reduced forms are inequivalent, and that every form is equivalent to a reduced form.

**Question 1** Find bounds for the coefficients of a reduced form of given discriminant and use these to write a procedure to list all the reduced forms with given discriminant  $d$ . Find all the reduced forms of discriminant  $d$  for  $-32 \leq d < 0$ , and indicate which of these forms are primitive.

The number of equivalence classes of primitive forms of discriminant  $d$  is the *class number*  $h(d)$ . This is equal to the number of primitive reduced forms. Sometimes a slightly different definition is used, without the requirement that the forms are primitive. However you should use the definition given here.

**Question 2** Tabulate both the number of reduced forms of discriminant  $d$ , and the class number  $h(d)$ , for  $-120 \leq d < 0$ . Comment on the relationship between these numbers. Also comment on the relationship between  $h(d)$  and  $h(dk^2)$ , when  $k$  is an odd prime (you may ignore  $d = -3, -4$  here). You may find it helpful to make a table with a large enough range of  $d$  and  $k$  to look for patterns.

### 3 Reduction of positive definite forms

We can find the reduced form equivalent to a given form  $f$  by *reduction*. If  $f$  is not reduced then  $c < a$ , or  $|b| > a$ , or  $a = -b$ , or  $a = c$  and  $b < 0$ . We define operations  $S$ ,  $T$  and  $T^{-1}$  on forms by

$$\begin{aligned} S: (a, b, c) &\mapsto (c, -b, a), \\ T: (a, b, c) &\mapsto (a, b + 2a, a + b + c), \\ T^{-1}: (a, b, c) &\mapsto (a, b - 2a, a - b + c). \end{aligned}$$

These operations are represented by matrices  $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$  and  $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$  in  $\text{SL}_2(\mathbb{Z})$ , so that each operation yields an equivalent form. If a form is not reduced then one of these operations may be applied and the result will be “closer” to a reduced form (in the sense that  $|a| + |b|$  is made smaller).

**Question 3** Write a program to find the reduced form equivalent to a given form. Your program should read in the coefficients  $a$ ,  $b$ ,  $c$  and print the reduced form equivalent to  $(a, b, c)$  together with the sequence of reduction operations which are needed. Run your program on the forms  $(220, 594, 401)$  and  $(226, 367, 149)$ .

### 4 Composition of forms

The *composition* of primitive forms  $f_1 = (a_1, b_1, c_1)$  and  $f_2 = (a_2, b_2, c_2)$ , with the same discriminant  $d$ , is defined as follows. First we put  $\beta = (b_1 + b_2)/2$  and  $\gamma = (b_2 - b_1)/2$ . Then we use Euclid’s algorithm twice. The first time we compute  $m = \text{gcd}(a_1, \beta)$  and find integers  $x$  and  $y$  with  $a_1x + \beta y = m$ . The second time we compute  $n = \text{gcd}(m, a_2)$  and solve the congruence

$$(m/n)z \equiv \gamma x - c_1 y \pmod{(a_2/n)}$$

for  $z$ . The composition of  $f_1$  and  $f_2$  is then

$$f_3 = f_1 \circ f_2 = (a_1 a_2 / n^2, b_1 + 2a_1 z / n, *)$$

where the third coefficient is chosen so that  $f_3$  also has discriminant  $d$ .

**Question 4** Write a program to compute the composition of two primitive forms. Briefly explain how you solve for  $z$ . It is known that if  $f_1 \sim g_1$  and  $f_2 \sim g_2$  then  $f_1 \circ f_2 \sim g_1 \circ g_2$ . As a way of testing your program, give some examples checking that this property holds.

Let  $d$  be a discriminant, *i.e.* a negative integer that is congruent to 0 or 1 modulo 4. It is known that the set of equivalence classes of primitive binary quadratic forms of discriminant  $d$  is an abelian group under composition. This is called the *class group*. The identity class contains either  $(1, 0, -d/4)$  or  $(1, 1, (1-d)/4)$ . The inverse of the class containing  $(a, b, c)$  is the class of  $(a, -b, c)$ .

It is known that every (non-trivial) finite abelian group may uniquely be written in the form

$$C_{n_1} \times C_{n_2} \times \dots \times C_{n_t}$$

where  $n_1, \dots, n_t$  are integers greater than one with  $n_1 | n_2 | \dots | n_t$ . One way to distinguish these groups is by counting the number of elements of each given order.

**Question 5** Determine the class group for all discriminants  $d$  between 0 and  $-120$ , and in addition for  $d = -48247$ ,  $-71411$  and  $-28959$ . You are not required to write a program that works for arbitrary  $d$ , but you are expected to explain your reasoning.

## 5 An application to factoring

For the remainder of this project we will only consider primitive forms.

A form  $(a, b, c)$  is *ambiguous* if it is equivalent to  $(a, -b, c)$ .

**Question 6** Find all reduced ambiguous forms of discriminant  $d$  for  $d = -240$ ,  $-627$  and  $-1428$ . Comment on the relationship between the reduced ambiguous forms of discriminant  $d$  and the factorisation of  $d$ . What do you notice about the number of such forms?

By *factoring* we mean the task of finding a non-trivial factor of a given composite integer  $N$ . The following method uses binary quadratic forms.

We take a discriminant  $d = -kN$ , with  $k$  a small positive integer, and attempt to construct ambiguous forms of discriminant  $d$ . To do this we pick a form at random and raise it to a suitable power in the class group. If this fails to produce an ambiguous form that factors  $N$ , we repeat with another randomly chosen form. One difficulty with this method is that it seems to require knowledge of the class number  $h(d)$ .

**Question 7** Explain, in terms of complexity, why computing  $h(d)$  using the methods in Section 2 would not lead to a useful factoring algorithm.

Instead we fix a positive integer  $B$  and assume that  $h(d)$  is a product of prime powers less than  $B$ . We choose a form of discriminant  $d$  at random (for example by choosing a small value of  $a$  at random, and then solving for  $b$  and  $c$  if possible) and successively raise it to each odd prime power less than  $B$ . We then repeatedly square this form in the hope of finding an ambiguous form that factors  $N$ . If this method repeatedly fails we might increase the value of  $B$ , or change the value of  $k$ .

**Question 8** Describe an efficient procedure for computing powers in the class group, based on the programs you wrote for Questions 3 and 4.

Illustrate the above method by using it to factor  $N = 12597203$ ,  $33377419$  and  $49047121$ . You should find it sufficient to work with  $k \leq 10$  and  $B \leq 50$ . In each case you should specify both the value of  $k$  and the sequence of forms computed.

## References

- [1] Buell, D. A. *Binary quadratic forms*.
- [2] Cassels, J. W. S. *Rational quadratic forms*.
- [3] Jones, B. W. *The arithmetic theory of quadratic forms*.
- [4] Koblitz, N. *A course in number theory and cryptography*.