二元二次型

在電腦上玩Gauss的實驗

于 靖

§1

整係數二元二次型是K.F.Gauss 1801年出版的Disquisitiones Arithmeticae（算學講話）的主要內容之一。以下我們要介紹的是如何在電腦上玩二元二次型。

首先，什麼是二元二次型（整係數）？

對固定之整數 a，b，c，二次齊次多項式

$$F=F(x,y)=ax^2+bxy+cy^2$$

稱爲二元二次型，以 $\{a,b,c\}$ 表示之。整數 b^2-4ac 稱爲此型之判別式。在本文裏我們只考慮所謂的原型，也就是 $\gcd(a,b,c)$ 總是爲 1 。

假使存在整數 x_0，y_0 使得

$$(*)\quad ax^2+bx_0y_0+cy_0^2=m,$$

m 爲給定整數，我們就說 $\{a,b,c\}$ 表示 m 。拿到二元二次型，就自然有下列問題：

⑴給定一個二次型，有那些整數可以被它表示？

⑵給定整數，那些二次型可以表示它？

⑶假使一個二次型可以表示某一整數，那麼會有多少種不同的表示法（也就是說方程式（＊）有多少個解）？

在回答上述問題時我們可以做變數變換：

$$x=\alpha x'+\beta y',\quad \alpha,\beta,\gamma,\delta\in Z$$
$$y=\gamma x'+\delta y',\quad \alpha\delta-\gamma\beta\neq 0 。$$

於是 $F(x,y)=ax^2+bxy+cy^2$ 就變成

$$F(x',y')=a'x'^2+b'x'y'+c'y'^2,$$

其中 $a'=a\alpha^2+b\alpha\gamma+c\gamma^2$，

$$b'=b(\alpha\delta+\beta\gamma)+2(a\alpha\beta+c\gamma\delta)$$
$$c'=a\beta^2+b\beta\delta+c\delta^2,$$

而另一方面判別式 $\triangle'=(\alpha\delta-\beta\gamma)^2\triangle$ 。假使 $\alpha\delta-\beta\gamma$ 是 ± 1，則變數變換可以返回來：

$$x'=\frac{\delta x-\beta y}{\alpha\delta-\beta\gamma},\quad y'=\frac{\alpha y-\gamma x}{\alpha\delta-\beta\gamma}。$$

我們只有興趣於 $\begin{vmatrix}\alpha & \beta\\ \gamma & \delta\end{vmatrix}=\pm 1$ 的情形，在此情形下就說 $\{a,b,c\}$ 和 $\{a',b',c'\}$ 是等價的。所有整係數相同判別式的二元二次型就分成了等價類。同類的二次型表示相同的整數。因而在考慮前述問題⑴，⑵時，等價類是很方便的觀念。問題⑶是一個很深奧的問題，（其最簡單的例子就是解整係數方程 $x^2+y^2=m$），在此我們限於篇幅不作進一步討論。

我們通常把二次型等價類再細分一點，用眞等價（Proper equivalence），也就是說只用行列式 $\begin{vmatrix}\alpha & \beta\\ \gamma & \delta\end{vmatrix}=1$ 的變數變換。下面我們講等價都是指眞等價。

假使判別式$\triangle = b^2 - 4ac < 0$，$a$與$c$必需同號，我們只考慮$a$與$c$均爲正數的情形，也就是所謂的正定型，正定型不能表示負整數。假使判別式$\triangle > 0$，則稱爲不定型。

拿到一個二次型，想知道它的類，我們就在每一類中找一個最簡單的（reduced）簡化型，然後用類似輾轉相除法把所給的二次型化簡（reduction）到簡化型。假如$\{a, b, c\}$是正定型，稱其爲簡化型就是指滿足$|b| \leq a \leq c$，若$a = c$或$a = |b|$則要要求$b > 0$。這樣一來給定判別式的正定型每一類有唯一的一個簡化型，因而類數是有限的，可以數出來。這些類數對應於虛二次數體的類數，是數論裏非常有意思的部份。Gauss猜到了只有以下9個（正定型）基本判別式的類數是1：

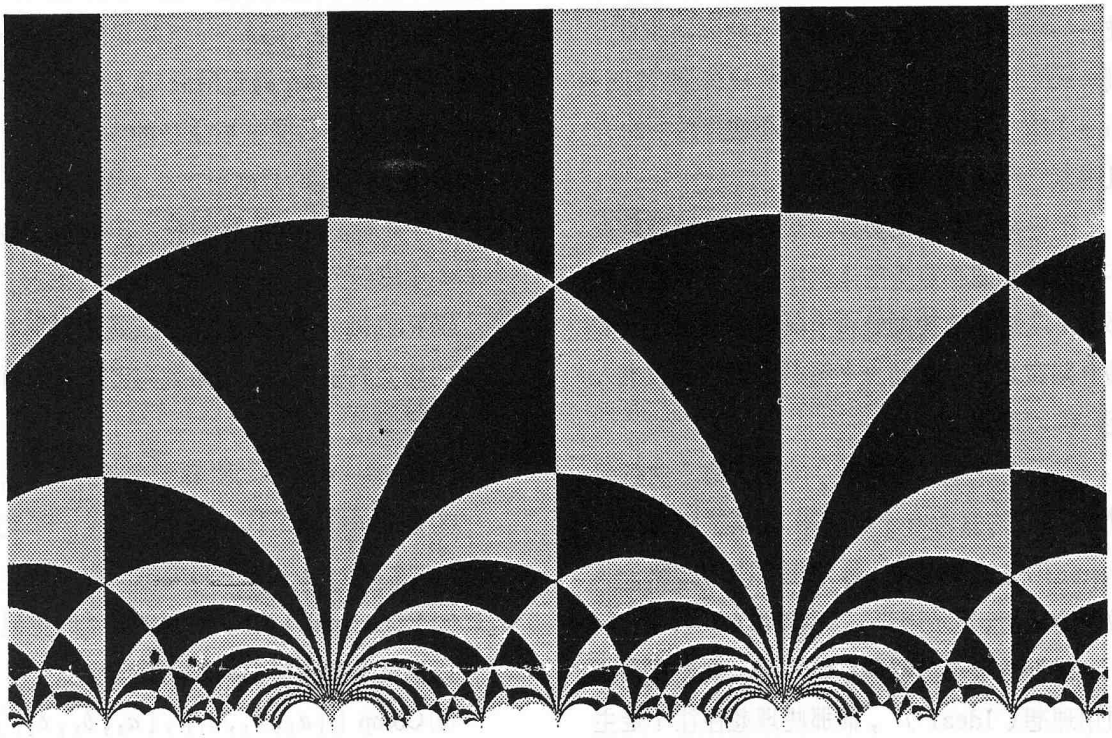
$$-3, -4, -7, -8, -11, -19,$$
$$-43, -67, -163$$

這個猜測到1960年代才被證明（K. Heegner, A. Baker, H. Stark）。

一個奇數判別式△假使無平方因數就稱爲基本判別式，對於偶數判別式而言，假使$\triangle/4$無平方因數而且$\triangle/4 \equiv 2$或$3 \pmod 4$，稱爲基本判別式。

更有意思的是Gauss發現二次型之間（判別式相同）有自然的運算，因而固定判別式的有限個類構成一個有限交換群。你可以算出這個群的構造。一般而言整個群算起來是很麻煩的，但是它的Sylow 2-子群卻比較好算。從有限群結構定理知道它總是可以分解成有限個循環2-子群的直積。假如所考慮判別式是基本判別式則這些 2-子群的個數正好等於判別式的不同質因數個數減1（正定型的情形，不定型情形可能再差1）。這個 2-子群的故事並不 trivial，也是Gauss發掘出來的。

正定型的化簡還有有意思的幾何解釋。由正定型$\{a, b, c\}$解二次方程式$aX^2 + bX + c = 0$，得一對虛根，在複數平面上取上半平面（虛部> 0）的“主根”，把上半平面看成非歐幾何平面，



（此圖是用Mathematica在電腦上畫的）

圖1

$$z \to \frac{\alpha z+\beta}{\gamma z+\delta}, \quad \begin{vmatrix} \alpha & \beta \\ \gamma & \delta \end{vmatrix} = 1,$$

$\alpha, \beta, \gamma, \delta \in Z$

看成非歐幾何平移。上圖中的非歐幾何三角形（一黑加一白）就可以經由所有的這些平移正好蓋住整個上半平面：（見上頁圖）化簡正定型的幾何意義就是經由 $\begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ 把二次型換成同等價類的型，而使其主根落在上半平面正中央，走向 $i\infty$ 的非歐幾何三角形（稱爲基本域）內，

$$|z| \geq 1, \quad -\frac{1}{2} \leq Rez \leq \frac{1}{2}。$$

不定型 $\{a, b, c\}$ 的簡化型規定如下：

$0 < b < \sqrt{\triangle}, \sqrt{\triangle} - b < 2|a| < \sqrt{\triangle} + b$。

固定判別式 $\triangle$，也只有有限個簡化型。不過落在同等價類的簡化型可以有許多。我們稱兩個簡化型 $\{a, b, c\}$ 與 $\{a', b', c'\}$ 是相接的，假使它們滿足 $c = a'$ 以及 $b + b' \equiv 0 \bmod (2a')$。每一個簡化正定型（判別式不爲平方）就會有唯一一個簡化正定型從右邊相接，也會有唯一一個簡化型從左邊相接。因此固定非平方判別式的所有簡化正定型就可分成各個（Cycles）輪。兩個同判別式的簡化型是等價的若且唯若它們在同一輪。

不定型的類數更奧妙，Gauss就問過是否存在無窮多個不定型基本判別式，其對應類數都是1？這相當於問是否有無窮多個實二次數體其類數是1。直到現在這還是數論裏沒有人能回答的問題。

二元二次型（整係數）的研究與二次數體 $Q(\sqrt{b})$ 的算術有密切關聯，其實唯有用Gauss的二次型理論，你才能實際的去做二次數體的算術，這是因爲，二次數環（如 $Z[\sqrt{-5}]$）一般不是唯一分解環（UFD），要做算術必須用理想（Ideal），而那些理想往往不是主理想（Principal ideal），因而不是自由模，要寫出所有這些理想類最容易的方式就是用對應的化簡二次型。所以二次型理論使得我們可以眞正去計算許多東西。在電腦上做實驗就很有意思了，讓你確實的“看”到數論裏一些著名定理及未解決的問題。

§ II

由於科技的進步及電腦的普及，對於現代喜歡「計算」的學生而言，在電腦上玩Gauss的數學實驗實在不是難事。目前市面上的一些數學計算軟體，如Mathematica，Maple，Macsyma等，都有很強的符號（代數）計算能力；運用這些軟體來寫程式玩二次型，甚至不需要先懂任何電腦語言。

我們的二次型程式（Binary DQF與Binary QF）是用Mathematica軟體（Wolfram公司，1991年2.0版）寫的。任何386/387以上的PC或68030以上的Macintosh電腦只要裝了這個軟體就可以跑下節裏所給的這兩套程式。在本節裏我們先把程式中寫進去函數作一介紹，然後附上一個Mathematica Session（Note Book），這是在Macintosh IIfx電腦上跑的紀錄（Mac IIfx的速度和一般386/387 PC是差不太多的）。

Binary DQF只跑正定型，而Binary QF是可處理正定型及不定型。Binary DQF中的主要部份是：

(1) Reduced Forms〔d〕，d 是判別式，給定判別式此一指令就列出所有的簡化型。

(2) Class No〔d〕，給判別式 d 的類數。

(3) Reduction〔$\{a, b, c\}$〕，給定二次型 $\{a, b, c\}$ 化簡它爲等價的簡化型。

(4) Principal〔d〕，給判別式 d 的主型，也就是對應類群的單位元素。

(5) Comp〔$\{a_1, b_1, c_1\}, \{a_2, b_2, c_2\}$〕，做兩個二次型的合成運算，我們的答案總是給簡化型。

(6) Sqa F〔{a,b,c}〕，做二次型平方運算，答案也總是給簡化型。

(7) Prin Root〔{a,b,c}〕，算對應二次方程式在上半平面的虛根。

(8) Fund SL2〔z〕，把複數上半平面的點，用行列式1的整係數線性變換（非歐幾何平移）搬到基本域裏（參考上節圖1）。

(9) Ambiguous〔d〕，列出所有判別式 d 的二次型等價類，其平方爲主類。也就是算出類群的 2-torsion 部份。

第二套 Binary QF 程式同時處理正定型及不定型，其中不同於第一套的部份包括：

(1) Class Reduced〔d〕，給定非平方判別式 $d>0$，列出所有的簡化型、輪，也就是列出所有的類。

(2) Adjacent〔{a,b,c}〕，給定簡化不定型，算出其相接的簡化型。

(3) Fcycles〔{a,b,c}〕，給定判別式不爲平方的簡化型，算出其簡化型所在的輪。

(4) Principal〔d〕，對於非平方判別式 $d>0$，給出類群單位元素所對應的輪。

總之，正定型我們用單個簡化型代表類，不定型則用簡化型所在的輪代表類。

```
<<NumberTheory`BinaryDQF`

ReducedForms[-383]//Timing
{8.55 Second, {{1, 1, 96}, {2, 1, 48}, {2, -1, 48}, {3, 1, 32},
   {3, -1, 32}, {4, 1, 24}, {4, -1, 24}, {6, 1, 16}, {6, -1, 16},
   {8, 1, 12}, {8, -1, 12}, {7, 3, 14}, {7, -3, 14}, {6, 5, 17},
   {6, -5, 17}, {9, 7, 12}, {9, -7, 12}}}
ClassNo[-383]
17
Reduction[{15, 20, 391}]//Timing
{0.366667 Second, {15, -10, 386}}
PrinRoot[{15, 20, 391}]
  2
-(-) + 5.06184 I
  3
FundSL2[PrinRoot[{15, 20, 391}]]//Timing
                  1
{0.316667 Second, - + 5.06184 I}
                  3
PrinRoot[{15, -10, 386}]
```

```
1
- + 5.06184 I
3

Table[{- i 4, Length[ReducedForms[- i 4]],
    ReducedForms[- i 4]}, {i, 1, 55}]//Timing

Table[{-3 - i 4, Length[ReducedForms[-3 - i 4]],
    ReducedForms[-3 - i 4]}, {i, 0, 40}]//Timing


Map[ClassNo[#]&,
{-3, -4, -7, -8, -11, -19, -43, -67, -163}]//Timing

{2.13333 Second, {1, 1, 1, 1, 1, 1, 1, 1, 1}}


ReducedForms[-1520]//Timing

{10.55 Second, {{1, 0, 380}, {4, 0, 95}, {5, 0, 76}, {19, 0, 20},

   {3, 2, 127}, {3, -2, 127}, {9, 8, 44}, {9, -8, 44}, {11, 8, 36},

   {11, -8, 36}, {12, 8, 33}, {12, -8, 33}, {15, 10, 27},

   {15, -10, 27}, {13, 12, 32}, {13, -12, 32}}}

Ambiguous[-1520]//Timing

{2.75 Second, {{1, 0, 380}, {4, 0, 95}, {5, 0, 76}, {19, 0, 20}}}

(*The following cell counts the number of distict
prime factors for even fundamental discriminants.*)

Table[If[Mod[m, 4] == 1 || Mod[m, 4] == 2,
     {- 4 m, 2^(Length[FactorInteger[4 m]] - 1)},
     {}], {m, 1, 55}]//Timing

(*The following cell computes the 2-torsion part
of the class groups for fundamental discriminants,
according to a theorem of Gauss this agrees with
the output of the previous cell.*)

Table[If[Mod[m, 4] == 1 || Mod[m, 4] == 2,
     {- m 4, Length[Ambiguous[- m 4]]}, {}],
     {m, 1, 55}]//Timing

{31.7167 Second, {{-4, 1}, {-8, 1}, {}, {}, {-20, 2}, {-24, 2}, {},

   {}, {-36, 2}, {-40, 2}, {}, {}, {-52, 2}, {-56, 2}, {}, {},

   {-68, 2}, {-72, 2}, {}, {}, {-84, 4}, {-88, 2}, {}, {},
```

```
   {-100, 2}, {-104, 2}, {}, {}, {-116, 2}, {-120, 4}, {}, {},
   {-132, 4}, {-136, 2}, {}, {}, {-148, 2}, {-152, 2}, {}, {},
   {-164, 2}, {-168, 4}, {}, {}, {-180, 4}, {-184, 2}, {}, {},
   {-196, 2}, {-200, 2}, {}, {}, {-212, 2}, {-216, 2}, {}}}
Principal[-7]
{1, 1, 2}

Comp[{9, 8, 44}, {5, 0, 76}]//Timing
{0.433333 Second, {9, -8, 44}}
<<NumberTheory`BinaryQF`
ReducedForms[4*513]//Timing
{47.6833 Second, {{-32, 30, 9}, {-32, 34, 7}, {-29, 14, 16},
   {-29, 44, 1}, {-27, 18, 16}, {-27, 36, 7}, {-19, 38, 8},
   {-16, 14, 29}, {-16, 18, 27}, {-9, 30, 32}, {-9, 42, 8},
   {-8, 38, 19}, {-8, 42, 9}, {-7, 34, 32}, {-7, 36, 27},
   {-1, 44, 29}, {1, 44, -29}, {7, 34, -32}, {7, 36, -27},
   {8, 38, -19}, {8, 42, -9}, {9, 30, -32}, {9, 42, -8},
   {16, 14, -29}, {16, 18, -27}, {19, 38, -8}, {27, 18, -16},
   {27, 36, -7}, {29, 14, -16}, {29, 44, -1}, {32, 30, -9},
   {32, 34, -7}}}
Principal[1173]//Timing
{1.03333 Second, {{-21, 9, 13}, {13, 17, -17}, {-17, 17, 13},
   {13, 9, -21}, {-21, 33, 1}, {1, 33, -21}}}
Reduction[{1, 1, -293}]
{1, 33, -21}
Table[{ i 4, ClassReduced[ i 4]}, {i, 1, 55}]//Timing

Table[{1 + i 4, ClassReduced[1 + 4 i]}, {i, 1, 55}]//Timing
```

```
ClassReduced[1173]//Timing
{18.15 Second, {4, {{7, 33, -3}, {-3, 33, 7}, {7, 23, -23},
    {-23, 23, 7}}, {{13, 17, -17}, {-17, 17, 13}, {13, 9, -21},
    {-21, 33, 1}, {1, 33, -21}, {-21, 9, 13}},
   {{21, 33, -1}, {-1, 33, 21}, {21, 9, -13}, {-13, 17, 17},
    {17, 17, -13}, {-13, 9, 21}},
   {{23, 23, -7}, {-7, 33, 3}, {3, 33, -7}, {-7, 23, 23}}}}
ClassReduced[1313]//Timing

Fcycle[{1, 4, -3}]//Timing
{0.283333 Second, {{-3, 2, 2}, {2, 2, -3}, {-3, 4, 1}, {1, 4, -3}}}
Ambiguous[1313]//Timing
{23.3833 Second, {{{14, 19, -17}, {-17, 15, 16}, {16, 17, -16},
    {-16, 15, 17}, {17, 19, -14}, {-14, 9, 22}, {22, 35, -1},
    {-1, 35, 22}, {22, 9, -14}, {-14, 19, 17}, {17, 15, -16},
    {-16, 17, 16}, {16, 15, -17}, {-17, 19, 14}, {14, 9, -22},
    {-22, 35, 1}, {1, 35, -22}, {-22, 9, 14}},
   {{13, 13, -22}, {-22, 31, 4}, {4, 33, -14}, {-14, 23, 14},
    {14, 33, -4}, {-4, 31, 22}, {22, 13, -13}, {-13, 13, 22},
    {22, 31, -4}, {-4, 33, 14}, {14, 23, -14}, {-14, 33, 4},
    {4, 31, -22}, {-22, 13, 13}}}}
```

Mathematica is a trademark of Wolfram Research Inc.

```
BeginPackage["BinaryDQF`"]

ReducedForms::usage ="List all reduced forms of a
```

```
given discriminant."
Euc2::usage ="a division algorithm"
Reduction::usage ="reduction of forms."
disc::usage ="discriminant"
Principal::usage ="principal form of given discriminant"
Comp::usage ="Composition of forms"
SqaF::usage ="Self composition"
Ambig::usage ="Self composition of all reduced forms"
Ambiguous::usage ="List all ambiguous classes."
PrinGenus::usage ="List all reduced forms in the same
genus as the principal form."
GenusF::usage ="List all reduced forms in the same
genus."
ClassNo::usage ="class number of given discriminant"
PrinRoot::usage ="Root in the upper half plane of the
quadratic equation"
FundSL2::usage ="Non-euclidean motion by SL2Z mapping
points in the upper half into the fundamental domain"

Begin["`Private`"]

Clear[ReducedForms, Reduction, disc, Comp, SqaF,
GenusF, FundSL2, ClassNo, PrinRoot, PrinGenus, Ambiguous,
Principal, Ambig, Euc2]

ReducedForms[d_Integer?Negative /; (Mod[d, 4] == 0 ||
Mod[d, 4] == 1)] := ReducedForms[d] =
  Module[{Answer, n, L, i, a, b, c}, Answer = {};
    For[n = Ceiling[-d/4], n <= -d/3 + 1, n++,
      L = Divisors[n];
      For[i = 1, 2i <= Length[L] + 1, i++,
        a = L[[i]];  c = n/a;  b = Sqrt[d + 4 a c];
        If[IntegerQ[b],
        If[c > a, If[b == a && GCD[c, a] ==1,
                  Answer = Append[Answer, {a, b, c}]];
                  If[b <  a && GCD[a, b, c] ==1,
                  Answer = Append[Answer, {a, b, c}];
                  If[b != 0 && GCD[a, b, c] ==1,
                  Answer = Append[Answer, {a,-b, c}]]]]];
        If[c == a, If[b <= a && GCD[a, b] ==1 ,
        Answer = Append[Answer, {a, b, c}]]]]];
        Answer]
ClassNo[d_Integer?Negative] := ClassNo[d]=
    Length[ReducedForms[d]]

Euc2[a_Integer, c_Integer] := Euc2[a, c] =
    If[Mod[a, 2 c] <= Abs[c], {(a - Mod[a, 2 c])/(2 c),
    Mod[a, 2 c]}, If[c > 0, {(a - Mod[a, 2 c])/(2 c) + 1,
```

```
     Mod[a, 2 c] - 2 c},
    {(a - Mod[a, 2 c])/(2 c) - 1, Mod[a, 2 c] + 2 c}]]

Reduction[f_List] := Reduction[f] =
    Module[{s = f},
    While[s[[1]] > s[[3]] || Abs[s[[2]]] > s[[1]],
    s ={s[[3]], -Euc2[s[[2]], s[[3]]][[2]], s[[1]] -
    s[[2]]*(Euc2[s[[2]], s[[3]]][[1]]) +
    s[[3]]*(Euc2[s[[2]], s[[3]]][[1]])^2}];
    If[(s[[1]] == s[[3]] || s[[1]] == Abs[s[[2]]]),
    {s[[1]], Abs[s[[2]]], s[[3]]}, s]]

disc[f_List] := disc[f] = f[[2]]^2 - 4 f[[1]] f[[3]]

Principal[d_Integer /; (Mod[d, 4] == 0 ||
    Mod[d, 4] == 1)] := Principal[d] =
    If[EvenQ[d], {1, 0, -d/4}, {1, 1, (1 - d)/4}]

Comp[f1_List, f2_List] := Comp[f1, f2] =
Module[{bb = (f1[[2]] + f2[[2]])/2, m, n, z, zz, A},
    m = ExtendedGCD[f1[[1]], bb];
    n = GCD[m[[1]], f2[[1]]];
    z = (m[[2, 1]]*(f2[[2]] - f1[[2]])/2 -
    f1[[3]]*m[[2, 2]])* PowerMod[m[[1]]/n, -1, f2[[1]]/n];
    zz = Mod[z, f2[[1]]/n];
    A ={f1[[1]] f2[[1]]/(n^2), f1[[2]] + 2 f1[[1]]* zz/n};
Reduction[{A[[1]], A[[2]], (A[[2]]^2 - disc[f1])/(4 A[[1]])}]]

SqaF[f_List] := SqaF[f] = Comp[f, f]

Ambig[d_Integer] := Ambig[d] =
 Map[{#, SqaF[#]}&, ReducedForms[d]]
Ambiguous[d_Integer] := Ambiguous[d] =
    Flatten[Table[If[Ambig[d][[i, 2, 1]] ==1,
    {Ambig[d][[i, 1]]}, {}], {i, 1, Length[Ambig[d]]}], 1]

PrinGenus[d_Integer] := PrinGenus[d] =
    Union@Map[SqaF[#]&, ReducedForms[d]]

GenusF[f_List] :=
    Prepend[Map[Comp[f, #]&, Rest[PrinGenus[disc[f]]]], f]

PrinRoot[f_List /; disc[f] < 0] := PrinRoot[f] =
    (-f[[2]] + I*N[Sqrt[-disc[f]]])/(2*f[[1]])

FundSL2[z_] := FundSL2[z] =
    Module[{s = z - Round[Re[z]]},
    While[N[Abs[s]] < 1, s = -1/s - Round[Re[-1/s]]];
    s]
```

```
End[]
EndPackage[]


Mathematica is a trademark of Wolfram Research Inc.


BeginPackage["BinaryQF`"]

ReducedForms::usage ="List all reduced forms of a
given discriminant."
Euc2::usage ="a division algorithm"
Reduction::usage ="reduction of forms."
disc::usage ="discriminant"
Principal::usage ="principal form of given discriminant"
Comp::usage ="Composition of forms"
SqaF::usage ="Self composition"
Ambig::usage ="Self composition of all reduced forms"
Ambiguous::usage ="List all ambiguous classes."
PrinGenus::usage ="List all reduced forms in the same
genus as the principal form."
GenusF::usage ="List all reduced forms in the same
genus."
ClassNo::usage ="class number of given discriminant"
Fcycle::usage="Find the cycle containing the reduction
of given form"
del::usage=""
ClassReduced::usage="List all redeuced forms arranged
in cycles"
Adjacent::usage ="Give the reduced form adjacent to the given
reduced form"

Begin["`Private`"]

Clear[ReducedForms, Reduction, disc, Comp, SqaF, GenusF, ClassNo,
PrinGenus, Ambiguous, Principal, Ambig, Euc2, ClassReduced,
Adjacent, del, Fcycle]

ReducedForms[d_Integer/; (Mod[d, 4] == 0 ||
Mod[d, 4] == 1)] := ReducedForms[d] =If[d > 0,
    Module[{Answer, n, L, i, a, b, c}, Answer = {};
    For[n = 1, n <=Floor[d/4] - 1, n++,
      L = Divisors[n];
      For[i = 1, 2 i <= Length[L] + 1, i++,
        a = L[[i]];  c = -n/a;  b = Sqrt[d + 4 a c];
      If[IntegerQ[b],
      If[N[Sqrt[d]] + b > 2 Abs[a],
```

```
   If[N[Sqrt[d]] - b < 2 Abs[a] && GCD[a, b, c] ==1,
   Answer = Union[Answer, {{a, b, c}, {-a, b, -c}, {c, b, a},
   {-c, b, -a}}]]]]]]; Answer],

Euc2[a_Integer, c_Integer] := Euc2[a, c] =
    If[Mod[a, 2 c] <= Abs[c], {(a - Mod[a, 2 c])/(2 c),
    Mod[a, 2 c]}, If[c > 0, {(a - Mod[a, 2 c])/(2 c) + 1,
     Mod[a, 2 c] - 2 c},
    {(a - Mod[a, 2 c])/(2 c) - 1, Mod[a, 2 c] + 2 c}]]

Reduction[f_List] := Reduction[f] =
    If[disc[f] < 0, Module[{s = f},
    While[s[[1]] > s[[3]] || Abs[s[[2]]] > s[[1]],
    s ={s[[3]], -Euc2[s[[2]], s[[3]]][[2]], s[[1]] -
    s[[2]]*(Euc2[s[[2]], s[[3]]][[1]]) +
    s[[3]]*(Euc2[s[[2]], s[[3]]][[1]])^2}];
    If[(s[[1]] == s[[3]] || s[[1]] == Abs[s[[2]]]),
    {s[[1]], Abs[s[[2]]], s[[3]]}, s]],
    Module[{s = f},
    While[((s[[2]] >= N[Sqrt[disc[s]]]) || (2Abs[s[[1]]] >=
    s[[2]] + N[Sqrt[disc[s]]]) || (2Abs[s[[1]]] <=
    N[Sqrt[disc[s]]] - s[[2]])),
    s ={s[[3]], -s[[2]] + 2 s[[3]]* del[s], s[[1]] -
    s[[2]]* del[s] +
    s[[3]]*(del[s])^2}]; s]]

Comp[f1_List, f2_List] := Comp[f1, f2] =
Module[{bb = (f1[[2]] + f2[[2]])/2, m, n, z, zz, A},
    m = ExtendedGCD[f1[[1]], bb];
    n = GCD[m[[1]], f2[[1]]];
    z = (m[[2, 1]]*(f2[[2]] - f1[[2]])/2 -
        f1[[3]]*m[[2, 2]])* PowerMod[m[[1]]/n, -1, f2[[1]]/n];
    zz = Mod[z, f2[[1]]/n];
    A ={f1[[1]] f2[[1]]/(n^2), f1[[2]] + 2 f1[[1]]* zz/n};
Reduction[{A[[1]], A[[2]], (A[[2]]^2 - disc[f1])/(4 A[[1]])}]]

SqaF[f_List] := SqaF[f] = Comp[f, f]

Principal[d_Integer /; (Mod[d, 4] == 0 ||
Mod[d, 4] == 1)] := Principal[d] =
If[d >0, Fcycle[Reduction[
If[EvenQ[d], {1, 0, -d/4}, {1, 1, (1 - d)/4}]]],
If[EvenQ[d], {1, 0, -d/4}, {1, 1, (1 - d)/4}]]

PrinGenus[d_Integer] := PrinGenus[d] =
    Union[Map[Sort[#]&, Map[Fcycle[SqaF[#]]&,
    Table[ClassReduced[d][[i, 1]],
    {i, 2, Length[ClassReduced[d]]}]]]]
```

```
    Module[{Answer, n, L, i, a, b, c}, Answer = {};
    For[n = Ceiling[-d/4], n <= -d/3 + 1, n++,
      L = Divisors[n];
      For[i = 1, 2i <= Length[L] + 1, i++,
        a = L[[i]];  c = n/a;  b = Sqrt[d + 4 a c];
        If[IntegerQ[b],
        If[c > a, If[b == a && GCD[c, a] ==1,
                  Answer = Append[Answer, {a, b, c}]];
                  If[b <  a && GCD[a, b, c] ==1,
                  Answer = Append[Answer, {a, b, c}];
                  If[b != 0 && GCD[a, b, c] ==1,
                  Answer = Append[Answer, {a,-b, c}]]]]];
        If[c == a, If[b <= a && GCD[a, b] ==1 ,
        Answer = Append[Answer, {a, b, c}]]]];
        Answer]]

ClassNo[d_Integer] := ClassNo[d]=
    If[d < 0, Length[ReducedForms[d]], ClassReduced[[1]]]

del[f_List] := del[f] =
    If[f[[3]] > 0, Floor[(N[Sqrt[disc[f]]] +
    f[[2]])/(2 f[[3]])],
Ceiling[(N[Sqrt[disc[f]]] + f[[2]])/(2 f[[3]])]]

Adjacent[f_List] := Adjacent[f] =
{(disc[f] -(-f[[2]] +
2 f[[1]]* del[{f[[3]], f[[2]], f[[1]]}])^2)/(-4 f[[1]]),
-f[[2]] + 2 f[[1]]* del[{f[[3]], f[[2]], f[[1]]}], f[[1]]}

Fcycle[f_List /; !IntegerQ[Sqrt[disc[f]]]] := Fcycle[f] =
Module[{Answer = {f}},
    While[!MemberQ[Answer,Adjacent[Answer[[1]]]],
    Answer = Prepend[Answer, Adjacent[Answer[[1]]]]];
    Answer]

ClassReduced[d_Integer?Positive /;(Mod[d, 4] == 0 ||
Mod[d, 4] == 1)] := ClassReduced[d] =
    If[IntegerQ[Sqrt[d]] || ReducedForms[d] == {} , {},
    Module[{Answer = {Fcycle[ReducedForms[d][[1]]]}},
    While[Length[ReducedForms[d]] > Length[Flatten[
    Answer, 1]],
    Answer = Append[Answer, Fcycle[(Complement[ReducedForms[d],
    Flatten[Answer, 1]])[[1]]]];
    Prepend[Answer, Length[Answer]]]]

disc[f_List] := disc[f] = f[[2]]^2 - 4 f[[1]] f[[3]]
```

```
GenusF[f_List] := GenusF[f] =
    Map[Fcycle[Comp[f, #]]&,
    Table[PrinGenus[disc[f]][[i, 1]],
    {i, 1, Length[PrinGenus[disc[f]]]}]]

Ambig[d_Integer] := Ambig[d] =
 Map[{#, SqaF[#]}&, ReducedForms[d]]

Ambiguous[d_Integer] := Ambiguous[d] =
    If[d < 0, Flatten[Table[If[Ambig[d][[i, 2, 1]] ==1,
    {Ambig[d][[i, 1]]}, {}], {i, 1, Length[Ambig[d]]}], 1],
    Flatten[Table[If[Length[Intersection[Map[Reverse[#]&,
    ClassReduced[d][[i]]], ClassReduced[d][[i]]]] == 0, {},
    {ClassReduced[d][[i]]}],
    {i, 2, Length[ClassReduced[d]]}], 1]]

End[]
EndPackage[]
```