RATIONAL POINTS ON CONICS

BY

ERIK HILLGARTER

Diplomarbeit

zur Erlangung des akademischen Grades

"Diplom-Ingenieur"

in der Studienrichtung

Technische Mathematik

Eingereicht von

Erik Hillgarter

Oktober 7, 1996

Angefertigt am
Research Institute for Symbolic Computation
Technisch-Naturwissenschaftliche Fakultät
Johannes Kepler Universität
4040 Linz, Austria

Eingereicht bei
Univ.-Doz. Dr. Franz Winkler

ACKNOWLEDGMENTS

I am grateful to my advisor Dr. Franz Winkler for offering the possibility of writing a diploma thesis and also for providing a topic/problem for it.

ABSTRACT

In order to parametrize an algebraic curve of genus zero, one usually faces the problem of finding rational points on it. This problem can be reduced to find rational points on a (birationally equivalent) conic. In this paper, we deal with a method of computing such a rational point on a conic from its defining equation (we are only interested in exact, i. e. symbolic solutions). The method will then be extended to work over the rational function field too. This problem arises in the parametrization of surfaces over Q.

Contents

1 **Introduction** 3

2 **Rational points on rational conics** 5

2.1 Problem specification and solution strategy 5

2.2 Simplification of the general conic equation 7

2.2.1 Parabolic case 8

2.2.2 Hyperbolic and elliptic case 11

2.2.3 Algorithm for the parabolic case 15

2.2.4 Algorithm for the hyperbolic/elliptic case 16

2.3 Solution for the hyperbolic/elliptic case :
The Legendre Theorem 19

2.3.1 A proof by Mordell 20

2.3.2 An algorithmic proof 24

2.3.3 An algorithm for solving the Legendre Equation 34

2.4 Real points on rational conics 38

2.4.1 Algorithm for the real case 40

2.5 Concluding Remarks 40

3 **Conics over Q(t)** 41

3.1 Analogies with the rational case 41

3.1.1 Quadratical residues in $Q[t]$ 48

3.2 Algorithms for $Q(t)$ 50

4 **Quadratic forms over arbitrary fields of characteristic $\neq 2$** **58**

4.1 Equivalence of quadratic forms 58

4.2 Representation of field elements 61

4.3 Binary quadratic forms 64

A **Some numbertheoretic supplements** **66**

A.1 The Legendre Symbol 66

A.2 A proof of the Legendre Theorem using
Minkowski's Lattice Point Theorem 69

B **Implementation and Examples** **72**

B.1 Information for the user 72

B.2 Some examples 76

B.3 The Maple^{TM} code 77

Chapter 1

Introduction

We consider here a subproblem of the so called parametrization problem (see e. g. [GEBAUER 91], [SENDRA, WINKLER 91], and [SENDRA, WINKLER 96]). The latter consists of computing a parametric representation of an implicitly given (plane) algebraic curve of genus zero. In order to tackle that problem algorithmically one faces the problem of finding rational points on such a curve. This problem can be reduced by using a theorem of Hilbert/Hurwitz that says that every plane (rational) algebraic curve of genus zero with degree $d \geq 4$ is birationally equivalent to a plane algebraic curve of genus zero with degree $d-2$ (which can be determined algorithmically). Hence, by an iterated application of the above theorem one will finally face a curve of degree 3 or 2 (depending on whether one started with odd or even degree). Now one can determine a rational point on *this* curve and then invert the birational transformations in order to arrive at a rational point of the original curve. Since every such curve of degree 3 has a twofold rational singularity, the problem is trivial for curves of odd degree. So the only remaining problem is to determine a rational point on a plane algebraic curve of degree 2, i.e. on a conic. This is our concerns.

First of all, we show in chapter 2 how to decide whether there is a rational point on a (rational) conic and - if possible - how to compute such a point. If there is no rational point on the conic we might content ourselves with determining a real point (which is a simpler problem). For achieving those goals we transform the defining equation of the conic to a quadratic form, the so called *Legendre Equation*, which can be solved by numbertheoretic methods.

In chapter 3, I extend this method to the case where the defining conic equation has rational functions (over Q) as coefficients and the goal is to find a rational function on this "conic". This problem arises in the context of parametrizing surfaces over Q. Especially, the following three problems are then solvable :

1. Consider a surface $F(x, y, t) = 0$, where $F \in Q[x, y, t]$ is of total degree 2 in x and y. Find a curve on $F = 0$ that intersects every horizontal plane (i. e. $z = const$) exactly once.

2. Parametrize a conic $f(x, y) = 0$ (where $f \in Q(t)[x, y]$) with rational functions in s and coefficients in $Q(t)$.

3. Parametrize a surface $F(x, y, t) = 0$ (where $F \in Q[x, y, t]$ is of total degree 2 in x and y) with rational functions in s and t.

Chapter 4 gives an overview of the (theoretical) situation of quadratic forms over arbitrary finite fields.

In the appendix the reader finds some numbertheoretic supplements as well as the Maple code of an implementation together with some examples produced with it.

The potential reader might note that chapter 3 depends heavily on chapter 2. In order to satisfy the needs of readers who only want to use the results, mathematical derivations appear in different sections (within a chapter) than algorithms.

Through the whole paper we denote variables by x, y, z (and also primed versions of them) and we denote rational respectively integer constants by a, b, c, d, e, f (and primed versions of them).

Chapter 2

Rational points on rational conics

2.1 Problem specification and solution strategy

In this section, we regard irreducible curves of degree two (so called "conics") with rational coefficients, i. e. a conic is defined by an irreducible polynomial $g \in Q[x, y]$ of degree two as the set $\{(\overline{x}, \overline{y}) \in \overline{Q}^2 \mid g(\overline{x}, \overline{y}) = 0\}$. In the sequel we refer to

$$g(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.1}$$

as the general conic equation (2.1). From a geometrical point of view, conics are those curves that result from cutting a circle cone with a plane. Let us first clarify the problem under consideration.

Definition 1 (rational point on a conic) *We call $(\overline{x}, \overline{y}) \in Q^2$ a **rational point** on the conic defined by (2.1) iff*

$$g(\overline{x}, \overline{y}) = 0. \blacksquare$$

By finding a rational point on the conic we understand the following.

<u>**Problem of finding rational points :**</u>

Given : a quadratic polynomial $g \in Q[x, y]$ defining a conic.

Decide : is there a rational point on the conic, i.e. does there exist $(\overline{x}, \overline{y}) \in \overline{Q}^2$ such that $g(\overline{x}, \overline{y}) = 0$?

Find : such a rational point, if there is one on the conic. ■

The following theorem shows us that the existence of one rational point on a conic implies that there are infinitely many rational points on it.

Theorem 1 *On a curve of order two with rational coefficients lie no or infinitely many rational points.*

Proof. Suppose we have $\overline{x}, \overline{y} \in Q$ such that $g(\overline{x}, \overline{y}) = 0$. Consider the line through $(\overline{x}, \overline{y})$ with rational direction vector $\begin{pmatrix} u & 1 \end{pmatrix}^T$ parametrized by $\begin{pmatrix} \overline{x} & \overline{y} \end{pmatrix}^T + t \begin{pmatrix} u & 1 \end{pmatrix}^T$. We claim that the second intersection point of the line and the conic is also a rational point (the first intersection point is $(\overline{x}, \overline{y})$, corresponding to $t = 0$).

$$\begin{aligned} g(\overline{x} + tu, \overline{y} + t) &= a(\overline{x} + tu)^2 + b(\overline{x} + tu)(\overline{y} + t) + c(\overline{y} + t)^2 + \\ &\quad + d(\overline{x} + tu) + e(\overline{y} + t) + f \overset{g(\overline{x}, \overline{y}) = 0}{=} \\ &= t^2(au^2 + bu + c) + t(du + e + 2au\overline{x} + b\overline{x} + b\overline{y}u + 2c\overline{y}). \end{aligned}$$

So the second intersection point corresponds (consider $g(\overline{x}+tu, \overline{y}+t) = 0$) to the rational parameter

$$\overline{t} = -\frac{du + e + 2au\overline{x} + b\overline{x} + b\overline{y}u + 2c\overline{y}}{au^2 + bu + c}.$$

There are clearly infinitely many ways to choose $u \in Q$ such that $\overline{t}$ represents a nontrivial rational number, giving rise to infinitely many rational points on the curve of the form

$(\bar{x} + \bar{t}u, \bar{y} + \bar{t})$. ■

We will see that it makes sense to distinguish between parabolas on the one hand and ellipses and hyperbolas on the other hand, since on a parabola, we are guaranteed to find one (and therefore infinitely many) rational point(s). The principal design of an algorithm for finding a rational point could be as follows.

ALGORITHM RATIONAL POINT

IN : quadratic polynomial $g(x,y) = ax^2 + bxy + cy^2 + dx + ey + f$ with rational coefficients.

OUT : Decision of existence of a rational point. A rational point if one exists.

1. Decide if g defines a parabola.
2. If g represents a parabola, compute a rational point on it.
   Return "There is a rational point" and return the point.
3. Decide whether g defines an irreducible curve. If not, return "Degenerate case".
4. Decide whether there is a rational point on the ellipse/hyperbola. If so, compute one and return "There is a rational point" and return the point.
   Otherwise return "No rational point".

2.2 Simplification of the general conic equation

In order to determine rational points, we will transform (2.1) by an affine change of coordinates to a more appropriate equation for which solution methods are available. In this section we give the transformations for the two cases parabola and ellipse/hyperbola and formulate the corresponding procedures in a PASCAL-like pseudocode.

2.2.1 Parabolic case

The following assumptions on the coefficients of (2.1) have to be made in order to guarantee that the curve is a parabola :

$$
\begin{aligned}
&b^2 = 4ac, && \text{(parabolic case)}\\
&(a,c) \neq (0,0), && \text{(g has degree two)}\\
&(a,d) \neq (0,0), && \text{(g does not depend only on y)}\\
&(c,e) \neq (0,0), && \text{(g does not depend only on x)}\\
&c \neq 0 \Rightarrow 2cd \neq be, && \text{(g does not define two lines)}\\
&a \neq 0 \Rightarrow 2ae \neq bd. && \text{(g does not define two lines)}
\end{aligned}
$$

The following transformations can be found in [KRAETZEL 81]. First of all, let us assume $c \neq 0$. Then we have $g(x,y) = 0$ iff $4cg(x,y) = 0$.

Lemma 2 (Transformed parabolic equation) *For g with $b^2 = 4ac$ and $c \neq 0$ we have*

$$4cg(x,y) = (bx + 2cy + e)^2 + d'x + f',$$

where $d' = 4cd - 2be$, $f' = 4cf - e^2$.

Proof.

$$
\begin{aligned}
& (bx + 2cy + e)^2 + d'x + f' \\
= \; & b^2x^2 + 4bcxy + 4c^2y^2 + 2bex + 4cey + e^2 + (4cd - 2be)x + (4cf - e^2) \overset{b^2=4ac}{=} \\
= \; & 4acx^2 + 4bcxy + 4c^2y^2 + 4cdx + 4cey + 4cf = \\
= \; & 4c(ax^2 + bxy + cy^2 + dx + ey + f) = 4cg(x,y). \blacksquare
\end{aligned}
$$

So far we have :

$$g(x,y) = 0 \text{ iff } (bx + 2cy + e)^2 + d'x + f' = 0.$$

At this stage, we might explain why the condition $2cd \neq be$ (i.e. $d' \neq 0$) was required : $(bx + 2cy + e)^2 + f' = 0$ is equivalent to $bx + 2cy + e = \pm\sqrt{-f'}$. Even if $\sqrt{-f'}$ is not complex, this equation just defines two parallel (real) lines.

Since we have $d' \neq 0$, a rational solution is given by

$$\overline{x} = -\frac{f'}{d'}, \overline{y} = -\frac{e + b\overline{x}}{2c}.$$

(One gets this solution by setting the terms inside and outside the brackets to zero separately).

Now the only remaining case to treat is the one when $c = 0$. Then - since we required $(a, c) \neq (0, 0)$ - we have $a \neq 0$. By interchanging the roles of x and y (or by considering $4ag(x, y) = 0$ and proceeding as above) we get (be aware of $\overline{x} \longleftrightarrow \overline{y}, a \longleftrightarrow c, d \longleftrightarrow e$) $d' = 4ae - 2bd$ and $f' = 4af - d^2$. Since $d' \neq 0$, a rational solution is given by

$$\overline{y} = -\frac{f'}{d'}, \overline{x} = -\frac{d + b\overline{y}}{2a}.$$

EXAMPLES

Example 1 *Consider $g(x, y) = x^2 + y$, i.e. $(a, b, c, d, e, f) = (1, 0, 0, 0, 1, 0)$.*[1]
Since $a \neq 0$ we get

$$d' = 4ae - 2bd = 4, \text{and}$$
$$f' = 4af - d^2 = 0.$$

So we get

$$y_1 = -\frac{f'}{d'} = 0, x_1 = -\frac{d + by_1}{2a} = -\frac{0 + 0}{2} = 0.$$

$(x_1, y_1) = (0, 0)$ is indeed a solution of $g(x, y) = x^2 + y = 0$.

[1] *Clearly $f = 0$ implies $g(0, 0) = 0$.*

Example 2 *Consider* $g(x,y) = y^2 + x + 1$, *i.e.* $(a,b,c,d,e,f) = (0,0,1,1,0,1)$.
Since $c \neq 0$ *we get*

$$d' = 4cd - 2be = 4, \text{ and}$$
$$f' = 4cf - e^2 = 4.$$

So we get

$$x_2 = -\frac{f'}{d'} = -1, y_2 = -\frac{e + bx_2}{2c} = -\frac{0+0}{2} = 0.$$

Indeed, $g(-1,0) = 0^2 + (-1) + 1 = 0$.

Example 3 *Consider* $g(x,y) = x^2 + 2xy + y^2 + x + 2y - 2$, *i.e.* $(a,b,c,d,e,f) = (1,2,1,1,2,-2)$.

Since $a \neq 0$ *and* $c \neq 0$ *we might use both formulae. Let us first of all use the formula for the case* $a \neq 0$.

$$d' = 4ae - 2bd = 8 - 4 = 4, \text{ and}$$
$$f' = 4af - d^2 = -8 - 1 = -9.$$

So we get

$$y_3 = -\frac{f'}{d'} = \frac{9}{4}, x_3 = -\frac{d + by_3}{2a} = -\frac{1+\frac{9}{2}}{2} = -\frac{11}{4}.$$

Indeed $g(-\frac{11}{4}, \frac{9}{4}) = 0$, *as one might check.*

Now we use the formula for the case $c \neq 0$:

$$d' = 4cd - 2be = 4 - 8 = -4, \text{ and}$$
$$f' = 4cf - e^2 = -8 - 4 = -12.$$

Hence we get

$$x_4 = -\frac{f'}{d'} = -\frac{12}{4} = -3, y_4 = -\frac{e + bx_4}{2c} = -\frac{2+2(-3)}{2} = 2.$$

Again, $g(-3,2)=0$ *holds.*

2.2.2 Hyperbolic and elliptic case

Again we consider (2.1), but we impose other conditions on the coefficients. We use again [KRAETZEL 81]. First of all, let

$$\begin{aligned} D &= 4ac-b^2,\\ N &= 4de-4bf,\\ M_1 &= 4c^2d^2-4bcde+4ace^2+4b^2cf-16ac^2f,\\ M_2 &= 4a^2e^2-4bade+4acd^2+4b^2af-16ca^2f. \end{aligned}$$

With these definitions we require

$D \neq 0,$	(hyperbolic/elliptic case)
$a=c=0 \Rightarrow N \neq 0,$	(g does not define two lines)
$c \neq 0 \Rightarrow M_1 \neq 0,$	(g does not define two lines)
$a \neq 0 \Rightarrow M_2 \neq 0,$	(g does not define two lines)
$(c \neq 0 \wedge D > 0) \Rightarrow M_1 > 0,$	(on the conic is more than one real point)
$(a \neq 0 \wedge D > 0) \Rightarrow M_2 > 0.$	(on the conic is more than one real point)

We consider two cases.

(CASE $a=c=0$**)** In this case we have $b \neq 0$ and $N \neq 0$. In the new coordinates

$$\begin{aligned} x' &= b(x+y)+d+e,\\ y' &= b(x-y)-d+e \end{aligned}$$

the equation $4bg(x,y)=0$ has the following form :

$$(x')^2-(y')^2=N.$$

Note that $N = 0$ would imply that we consider the two lines $x' = y'$ and $x' = -y'$.
(CASE $c \neq 0$) We have $M_1 \neq 0$ and ($D > 0 \Rightarrow M_1 > 0$). Under the coordinate change

$$x' = Dx + 2dc - be,$$
$$y' = bx + 2cy + e$$

the equation $4cDg(x, y) = 0$ becomes

$$(x')^2 + D(y')^2 = M_1.$$

Note that $M_1 = 0$ would imply that we consider the two (possibly complex) lines $x' = \pm\sqrt{D}y'$.

Remark 1 *The case $a \neq 0$ is totally analogous to the case $c \neq 0$ (just interchange the roles of x and y and therefore also those of a and c respectively of d and e; in addition use M_2 instead of M_1).*

Proof. We let Maple™ simplify the considered equations.
(CASE $a = c = 0$)

$$\frac{(b(x+y)+d+e)^2 - (b(x-y)-d+e)^2 - (4de - 4bf)}{4b}$$

$$= bxy + xd + ye - f \stackrel{a=c=0}{=} g(x, y).$$

(CASE $c \neq 0$)

$$\frac{(x')^2 + D(y')^2 - M_1}{4cD}$$

$$= x^2a + y^2c + f + xd + bxy + ye = g(x, y). \blacksquare$$

In both cases we arrived at an equation of the form

$$X^2 + KY^2 = L, \tag{2.2}$$

where $K, L \in Q$, and in both cases we do not have $(K > 0 \wedge L < 0)$, which would exclude the existence of a real solution.

So let us now consider equations of this form . Switching to homogeneous coordinates we set

$$X = \frac{x}{z},\ Y = \frac{y}{z},\ K = \frac{b'}{a'},\ L = -\frac{c'}{a'}.$$

Note that if $K = k_1/k_2$, $L = l_1/l_2$ we may choose $a' = \operatorname{lcm}(k_2, l_2)$, $b' = k_1 l_2 / \gcd(k_2, l_2)$, and $c' = -l_1 k_2 / \gcd(k_2, l_2)$. Then (2.2) becomes the Diophantine equation

$$a'x^2 + b'y^2 + c'z^2 = 0. \tag{2.3}$$

Clearly a', b', and c' are nonzero and do not all have the same sign (look at their definitions and use $\neg(K > 0 \wedge L > 0)$). But we want to achieve more, namely the reduction of (2.3) to an equation of similar form whose coefficients are squarefree and pairwise relatively prime. We use ideas from [ROSE 88]. Let us assume that

$$a' = a_1' r_1^2,\ b' = b_1' r_2^2,\ c' = c_1' r_3^2,$$

where a_1', b_1', and c_1' are squarefree. Consider

$$a_1'x^2 + b_1'y^2 + c_1'z^2 = 0. \tag{2.4}$$

(2.4) has an integral solution iff (2.3) has one. For showing the nontrivial direction, assume that (2.4) has the integral solution $(\overline{x}, \overline{y}, \overline{z})$. Then

$$a'\left(\frac{\overline{x}}{r_1}\right)^2 + b'\left(\frac{\overline{y}}{r_2}\right)^2 + c'\left(\frac{\overline{z}}{r_3}\right)^2 = 0, \text{i. e.}$$

$$a'\,(\overline{x}\,r_2\,r_3)^2 + b'\,(\overline{y}\,r_1\,r_3)^2 + c'\,(\overline{z}\,r_1\,r_2)^2 \;=\; 0,$$

giving an integral solution of (2.3).

Remark 2 *In the end, we are only interested in the dehomogenization, so the rational solution* $(\overline{x}/r_1, \overline{y}/r_2, \overline{z}/r_3)$ *is enough for our purposes.*

Now, we divide (2.4) by $\gcd(a_1', b_1', c_1')$, getting

$$a''x^2 + b''y^2 + c''z^2 = 0. \tag{2.5}$$

What remains is to make the coefficients pairwise relatively prime.
Let $g_1 = \gcd(a'', b'')$, $a''' = a''/g_1$, $b''' = b''/g_1$, and let $(\overline{x}, \overline{y}, \overline{z})$ be an integral solution of (2.5). Then $g_1 \mid c''\,\overline{z}^2$, and hence, since $\gcd(a'', b'', c'') = 1$, we have $g_1 \mid \overline{z}^2$. Furthermore, since g_1 is squarefree (since a'', b'' are), we have $g_1 \mid \overline{z}$. So, letting $z = g_1 z'$ and cancelling (2.5) by g_1, we arrive at

$$a'''x^2 + b'''y^2 + \underbrace{c''g_1}_{c'''}\,(z')^2 = 0. \tag{2.6}$$

We have $\gcd(a''', b''') = 1$ and c''' is squarefree since g_1 and c'' are relatively prime. Repeating this process with $g_2 = \gcd(a''', c''')$ and $y = g_2\,y'$ we arrive at

$$a''''x^2 + \underbrace{b'''g_2}_{b''''}\,(y')^2 + c''''(z')^2 = 0. \tag{2.7}$$

(Again $a'''' = a'''/g_2$, $c'''' = c'''/g_2$).
Now we do it a last time with $g_3 = \gcd(b'''', c'''')$ and $x = g_3\,x'$. Let $a = a''''g_3$, $b = b''''/g_3$, and $c = c''''/g_3$. Then we arrive at

$$a(x')^2 + b(y')^2 + c(z')^2 = 0, \tag{2.8}$$

the so called Legendre Equation. We note : a, b, and c are nonzero, do not all have the same sign, are squarefree, and pairwise relatively prime. We will treat this equation in section 2.3.

2.2.3 Algorithm for the parabolic case

We use the results gained in subsection 2.2.1 for an algorithmic solution formulated in pseudocode.

PROC PARABOLA($\downarrow g \uparrow ok \uparrow \overline{x} \uparrow \overline{y}$)

IN:

$g \in Q[x, y]$, $g(x, y) = ax^2 + bxy + cy^2 + dx + ey + f$.

OUT:

ok : boolean.

$(ok = true)$ iff $g(x, y) = 0$ defines an irreducible parabola.

$\overline{x}$, $\overline{y} \in Q$.

$(ok = true)$ implies $g(\overline{x}, \overline{y}) = 0$.

LOCAL:

d', $f' \in Q$.

BEGIN

$ok := ((a, d) \neq (0, 0)) \wedge ((c, e) \neq (0, 0)) \wedge (b^2 = 4ac) \wedge ((a, c) \neq (0, 0))$;

if not ok **then** return;

if $(a \neq 0)$ **then**

  $(d', f') := (4ae - 2bd, 4af - d^2)$;

  if $(d' \neq 0)$ **then**

    $\overline{y} := -f'/d'; \overline{x} := -(d + b\overline{y})/2a$

  else

    $ok := false$

  end if

else # $(c \neq 0)$

  $(d', f') := (4cd - 2be, 4cf - e^2)$;

  if $(d' \neq 0)$ **then**

$\overline{x} := -f'/d'; \overline{y} := -(e + b\overline{x})/2c$

else

$ok := false$

end if

end if

END PARABOLA.

2.2.4 Algorithm for the hyperbolic/elliptic case

Here we formulate the knowledge from subsection 2.2.2 in algorithmic form. We assume the procedures *numer* and *denom*, which deliver numerator and denominator of a rational number. In addition, *sqfrp* should deliver the squarefree part of an integer, i.e. for $n = \prod_{p\ prime} p^{n_p}$ we have

$$sqfrp(n) = \prod_{p\ prime} p^{\mathrm{mod}(n_p,2)}.$$

Furthermore, we assume the procedure *Legendre* (presented in subsection 2.3.3) that decides whether the Legendre Equation has (nontrivial) integral solutions and eventually computes one (with $z \neq 0$). Also the theory of section 2.4 (computing a real point on the conic in case no rational one exists) is already used here.

PROC CONIC2($\downarrow a \downarrow b \downarrow c \downarrow d \downarrow e \downarrow f \uparrow ok \uparrow ratpoint \uparrow X \uparrow Y$)

IN:

a, b, c, d, e, $f \in Q$ defining the conic.

OUT:

ok, $ratpoint$: boolean.

$X, Y \in R$.

($ok = false$) means that we consider either a parabola or two lines, or that the conic has not more than one real point.

($ok = ratpoint = true$) implies that (X, Y) are coordinates of a rational point on the conic.

($ok = true$; $ratpoint = false$) implies that there is no rational point on the conic. In this case (X, Y) are coordinates of a real point on the conic.

LOCAL

$D,\ K,\ L \in Q$;

$k_1, k_2, l_1, l_2, g, a_1, a_2, b_1, b_2, c_1, c_2, r_1.r_2, r_3, g_1, g_2, g_3, x, y, z \in Z$.

BEGIN

$D := 4ac - b^2;\ ok := D \neq 0;$

if not ok **then** return;

if $a = 0$ and $c = 0$ **then**

  $K := -1;\ L := 4(de - bf)$

elseif $c \neq 0$ **then**

  $K := D;\ L := 4c^2d^2 - 4bcde + 4ace^2 + 4b^2cf - 16ac^2f$

else # $(a \neq 0 \wedge c = 0)$

  $K := D;\ L := 4a^2e^2 - 4bade + 4b^2af$

end if

$ok := L \neq 0 \wedge \neg(K > 0 \wedge L < 0);$

if not ok **then** return;

$k_1 := numer(K);\ k_2 := denom(K);$

$l_1 := numer(L);\ l_2 := denom(L);$

$g := \gcd(k_2, l_2);$

$a_1 := l_2k_2/g;\ b_1 := k_1l_2/g;\ c_1 := -l_1k_2/g;$

$a_2 := sqfrp(a_1);\ r_1 := sqrt(a_1/a_2);$

$b_2 := sqfrp(b);\ r_2 := sqrt(b_1/b_2);$

$c_2 := sqfrp(c);\ r_3 := sqrt(c_1/c_2);$

$g := \gcd(a_2, b_2, c_2);$

$a_2 := a_2/g;\ b_2 := b_2/g;\ c_2 := c_2/g;$

$g_1 := \gcd(a_2, b_2);$

$a_2 := a_2/g_1;\ b_2 := b_2/g_1;\ c_2 := c_2 g_1;$

$g_2 := \gcd(a_2, c_2);$

$a_2 := a_2/g_2;\ b_2 := b_2 g_2;\ c_2 := c_2/g_2;$

$g_3 := \gcd(b_2, c_2);$

$a_2 := a_2 g_3;\ b_2 := b_2/g_3;\ c_2 := c_2/g_3;$

CALL $Legendre(\downarrow a_2, \downarrow b_2, \downarrow c_2, \uparrow ratpoint, \uparrow x, \uparrow y, \uparrow z);$

if not *ratpoint* **then**

  if $L > 0$ **then**

    $x := sqrt(L);\ y := 0$

  else

    $x := 0;\ y := \sqrt{L/K}$

  end if

else

  $x := xg_3/r_1;\ y := yg_2/r_2;\ z := zg_1/r_3;$

  $x := x/z;\ y := y/z$

end if

if $a = 0$ and $c = 0$ **then**

  $X := (x + y - 2e)/2b;\ Y := (x - y - 2d)/2b$

elseif $c \neq 0$ **then**

  $X := (x - 2dc + be)/K;\ Y := (y - bX - e)/2c$

else # $(a \neq 0 \wedge c = 0)$

  $Y := (x - 2ea + bd)/K;\ X := (y - bY - d)/2a$

end if

END CONIC2

2.3 Solution for the hyperbolic/elliptic case : The Legendre Theorem

In subsection 2.2.2 we saw that the problem of finding a rational point on an ellipse/hyperbola reduces to the problem of finding a nontrivial integral solution of the so called Legendre Equation

$$ax^2 + by^2 + cz^2 = 0, \tag{2.9}$$

where a, b, and c are integers such that $abc \neq 0$. When speaking of a nontrivial integral solution we will always mean the following.

Definition 2 (Nontrivial Solution of LE) *We call* $(\overline{x}, \overline{y}, \overline{z}) \in Z^3$ *a nontrivial integral solution of (2.9) iff*

$$(\overline{x}, \overline{y}, \overline{z}) \neq (0,0,0) \text{ and } \gcd(\overline{x}, \overline{y}, \overline{z}) = 1. \blacksquare$$

We also pointed out that we may w. l. o. g. assume

$$\begin{aligned} a &> 0,\ b < 0, \text{ and } c < 0, && (2.10)\\ & a,\ b, \text{ and } c \text{ are } squarefree, && (2.11)\\ \gcd(a,b) &= \gcd(a,c) = \gcd(b,c) = 1. && (2.12) \end{aligned}$$

In this section we deal with necessary and sufficient conditions in order that (2.9) has nontrivial integral solutions. Such conditions are given by the Theorem of Legendre. We give Mordell's proof of it (see [ROSE 88]), but also a constructive one (by [IRELAND, ROSEN 82]) from which we will extract the algorithm given in subsection 2.3.3. For a formulation of Legendre's Theorem we need the notion of quadratical residues.

Definition 3 (Quadratical Residue) *Let m, n be nonzero integers. Then m is a quadratical residue modulo n (written $m\,R\,n$) iff*

$$\exists x \in Z : x^2 \equiv_n m. \blacksquare$$

Now we can state the theorem.

Theorem 3 (Legendre) *Suppose a, b, and c satisfy (2.10), (2.11) and (2.12). Then (2.9) has a nontrivial integral solution iff*

$$-ab\,R\,c, \quad -bc\,R\,a, \quad \textit{and} \quad -ac\,R\,b. \tag{2.13}$$

2.3.1 A proof by Mordell

We require two Lemmata.

Lemma 4 *Let n be a positive integer. Suppose α, β, and γ are positive irrational numbers whose product $\alpha\beta\gamma = n$. Then for every triple $(a_1, a_2, a_3) \in Z^3$, the congruence*

$$a_1 x + a_2 y + a_3 z \equiv_n 0$$

has a solution $(\overline{x}, \overline{y}, \overline{z}) \neq (0, 0, 0)$ which satisfies

$$|\overline{x}| < \alpha, \; |\overline{y}| < \beta, \; \textit{and} \; |\overline{z}| < \gamma.$$

Proof. Consider the set

$$S = \{(x, y, z) \in N_0^3 \mid x \leq \lfloor\alpha\rfloor \wedge y \leq \lfloor\beta\rfloor \wedge z \leq \lfloor\gamma\rfloor\}.$$

This set contains $(1+\lfloor\alpha\rfloor)(1+\lfloor\beta\rfloor)\ (1+\lfloor\gamma\rfloor) > \alpha\beta\gamma = n$ elements. But there are at most n residue classes modulo n, and so triples (x_1, y_1, z_1) and (x_2, y_2, z_2) occur in S satisfying

$$a_1x_1 + a_2y_1 + a_3z_1 \equiv_n a_1x_2 + a_2y_2 + a_3z_2.$$

The result follows if we take $\overline{x} = x_2 - x_1$, $\overline{y} = y_2 - y_1$, and $\overline{z} = z_2 - z_1$. ■

Lemma 5 *Let $m, n \in N$ with $\gcd(m, n) = 1$. Suppose the form $ax^2 + by^2 + cz^2$ can be expressed as a product of linear factors both modulo m and modulo n. Then it can be expressed as a product of linear factors modulo mn.*

Proof. Let the conditions be expressed by

$$\begin{aligned} ax^2 + by^2 + cz^2 &\equiv (a_1x + a_2y + a_3z)(a_4x + a_5y + a_6z) \pmod{m}, \\ ax^2 + by^2 + cz^2 &\equiv (a_1'x + a_2'y + a_3'z)(a_4'x + a_5'y + a_6'z) \pmod{m}. \end{aligned}$$

By the Chinese remainder theorem[2] we can find integers a_i^* satisfying $a_i^* \equiv_m a_i$ and $a_i^* \equiv_n a_i'$, for $i \in \{1,..., 6\}$. Combining these congruences we have

$$ax^2 + by^2 + cz^2 \equiv_{mn} (a_1^*x + a_2^*y + a_3^*z)(a_4^*x + a_5^*y + a_6^*z). ■$$

Now we proof Legendre's Theorem.

Proof. (Legendre's Theorem)
We first show that the conditions (2.13) are necessary. Let $(\overline{x}, \overline{y}, \overline{z})$ be a solution of (2.9); it follows that $\gcd(c, \overline{x}) = 1$. For if any prime p divides $\gcd(c, \overline{x})$, then p divides $b\overline{y}^2$ but

[2] This theorem states : Let m_1, m_2,..., m_k be pairwise relatively prime integers > 1, and let $M = m_1m_2...m_k$. Then there exists a unique nonnegative solution modulo M of the simultaneous congruences

$$x \equiv_{m_1} a_1, x \equiv_{m_2} a_2, ..., x \equiv_{m_k} a_k. \quad (a_i \in Z)$$

p does not divide b (since $\gcd(b, c) = 1$ by (2.12)) and so p divides $\overline{y}$. Consequently we have p^2 divides $a\overline{x}^2 + b\overline{y}^2$ and hence p^2 divides $c\overline{z}^2$. But c is squarefree and so p divides $\overline{z}$. This contradicts the assumption $\gcd(\overline{x}, \overline{y}, \overline{z}) = 1$.

As $\gcd(c, \overline{x}) = 1$ we can find $\overline{x}'$ satisfying $\overline{x}\overline{x}' \equiv_c 1$. Also, clearly

$$a\overline{x}^2 + b\overline{y}^2 \equiv_c 0,$$

and so, by multiplying with $b(\overline{x}')^2$,

$$b^2(\overline{x}')^2\overline{y}^2 \equiv_c -ab(\overline{x}\overline{x}')^2 \equiv_c -ab.$$

Thus $-ab\, R\, c$ holds. The remaining conditions can be derived similarly.

For proving the reverse implication we deal first with three special cases.

(Case b = c = -1) In this case (2.13) gives $-1\, R\, a$ and so, integers r and s exist satisfying $r^2 + s^2 = a$ (a constructive proof of this fact will be given in section 2.3.2). Hence in this case (2.9) has the solution $(\overline{x}, \overline{y}, \overline{z}) = (1, r, s)$.

(Case a = 1, b = -1) Here (2.9) has the solution $(\overline{x}, \overline{y}, \overline{z}) = (1, 1, 0)$.

(Case a = 1, c = -1) Here (2.9) has the solution $(\overline{x}, \overline{y}, \overline{z}) = (1, 0, 1)$.

In the general case we have $-ab\, R\, c$, that is an integer t can be found to satisfy

$$t^2 \equiv_c -ab. \tag{2.14}$$

Also (since $\gcd(a, c) = 1$ by (2.12)) a^* exists satisfying $aa^* \equiv_c 1$. Thus working modulo c we have

$$\begin{aligned} ax^2 + by^2 + cz^2 &\equiv aa^*(ax^2 + by^2) \equiv a^*(a^2x^2 + aby^2) \\ &\equiv a^*(a^2x^2 - ty^2) \equiv a^*(ax - ty)(ax + ty) \\ &\equiv (x - a^*ty)(ax + ty) \pmod{c}. \end{aligned}$$

Using the remaining conditions (2.13) we see that $ax^2+by^2+cz^2$ can also be expressed as a product of linear factors modulo b and modulo a and so, by Lemma 5, integers $a_1, \ldots, a_6$ can be found to satisfy

$$ax^2 + by^2 + cz^2 \equiv_{abc} (a_1x + a_2y + a_3z)(a_4x + a_5y + a_6z). \tag{2.15}$$

Note that this holds for all x, y and z. For the next part we consider the congruence

$$(a_1x + a_2y + a_3z) \equiv_{abc} 0. \tag{2.16}$$

As we have dealt with three special cases above, and as a, b and c satisfy (2.11) and (2.12), we may assume that $\sqrt{bc}$, $\sqrt{-ac}$, and $\sqrt{-ab}$ are irrational. Applying Lemma 4 to (2.16), with $\alpha = \sqrt{bc}$, $\beta = \sqrt{-ac}$, and $\gamma = \sqrt{-ab}$, integers x_1, y_1, and z_1 can be found to satisfy $(x_1, y_1, z_1) \neq (0,0,0)$, $a_1x_1 + a_2y_1 + a_3z_1 \equiv_{abc} 0$, and

$$|x_1| < \sqrt{bc},\ |y_1| < \sqrt{-ac}, \text{ and } |z_1| < \sqrt{-ab}. \tag{2.17}$$

Now combining (2.15) and (2.17) we have

$$ax_1^2 + by_1^2 + cz_1^2 \equiv_{abc} 0.$$

But, as b and c are negative, (2.17) also gives

$$ax_1^2 + by_1^2 + cz_1^2 \leq ax_1^2 < abc, \tag{2.18}$$

and, as a is positive,

$$\begin{aligned} ax_1^2 + by_1^2 + cz_1^2 &\geq by_1^2 + cz_1^2 > \\ b(-ac) + c(-ab) &= -2abc. \end{aligned} \tag{2.19}$$

These three relations (2.16), (2.18), and (2.19) imply that

$$\begin{aligned} ax_1^2 + by_1^2 + cz_1^2 &= 0, \text{ or} \\ ax_1^2 + by_1^2 + cz_1^2 &= -abc. \end{aligned}$$

If the first case holds the result follows, so we may assume that the second case holds. Let

$$x_2 = x_1 z_1 - by_1,\ y_2 = y_1 z_1 + ax_1,\ z_2 = z_1^2 + ab.$$

This gives

$$\begin{aligned} ax_2^2 + by_2^2 + cz_2^2 &= a(x_1 z_1 - by_1)^2 + b(y_1 z_1 + ax_1)^2 + c(z_1^2 + ab)^2 = \\ &= (ax_1^2 + by_1^2 + cz_1^2)z_1^2 - 2abx_1y_1z_1 + 2abx_1y_1z_1 + \\ &\quad + ab(by_1^2 + ax_1^2 + cz_1^2) + abcz_1^2 + a^2b^2c \\ &= (-abc)z_1^2 + ab(-abc) + abcz_1^2 + a^2b^2c = 0, \end{aligned}$$

using our assumption. This is a nontrivial solution. For if $z_1^2 + ab = 0$ then $a = 1$ and $b = -1$ as a and b are coprime and squarefree, but this case has been dealt with previously (see above). Thus nontrivial solutions have been found in all cases and the proof is complete.

2.3.2 An algorithmic proof

Now we present an algorithmic proof of the Legendre Theorem that gives immediately an algorithm for finding a solution of (2.9) if one exists (see section 2.3.3). We follow [IRELAND, ROSEN 82]. First of all let us state the Legendre Theorem again.

Theorem 6 (Legendre, Version 1) *Let a, b, c be nonzero integers, squarefree, pairwise relatively prime and not all positive nor all negative. Then*

$$ax^2 + by^2 + cz^2 = 0 \tag{2.20}$$

has a nontrivial integral solution iff the following conditions are satisfied.

$$-ab\,R\,c, \tag{2.21}$$

$$-ac\,R\,b, \tag{2.22}$$

$$-bc\,R\,a. \tag{2.23}$$

■

We prove this result in the following equivalent form.

Theorem 7 (Legendre, Version 2) *Let a and b be positive squarefree integers. Then*

$$ax^2 + by^2 = z^2 \tag{2.24}$$

has a nontrivial solution iff the following three conditions are satisfied.

$$a\,R\,b, \tag{2.25}$$

$$b\,R\,a, \tag{2.26}$$

$$-\frac{ab}{\gcd(a,b)^2}\,R\,\gcd(a,b). \tag{2.27}$$

Proof. (Equivalence of Theorem 6 and Theorem 7)

(Version 1 implies Version 2) Consider

$$ax^2 + by^2 = z^2 \tag{2.28}$$

as in Theorem 7. Let $g = \gcd(a, b)$, $a' = a/g$, $b' = b/g$. We know already (compare subsection 2.2.2) that (2.28) has a nontrivial integral solution iff

$$a'x^2 + b'y^2 - gz^2 = 0 \tag{2.29}$$

has one. Clearly, a', b', and $-g$ are nonzero integers, squarefree, pairwise relatively prime and, not all positive nor all negative. Hence by Theorem 6, (2.29) has a nontrivial integral solution iff

$$-a'b' \, R - g, \tag{2.30}$$

$$-a'(-g) \, R \, b', \tag{2.31}$$

$$-b'(-g) \, R \, a' \tag{2.32}$$

are satisfied. (2.30)-(2.32) can be written as

$$\frac{-ab}{g^2} \, R \, g, \tag{2.33}$$

$$a \, R \, b', \tag{2.34}$$

$$b \, R \, a'. \tag{2.35}$$

But (2.33) already gives (2.27). By (2.34) and $a \, R \, g$ we get by Lemma 9 (given after the proof of Theorem 7) $a \, R \, b$, i. e. we get (2.25). By (2.35) and $b \, R \, g$ we get by Lemma 9 $b \, R \, a$, i. e. we get (2.26).

(Version 2 implies Version 1) We assume Theorem 7 and consider (2.20) with a, b, and c as in Theorem 6. Let us assume that a and b are positive while c is negative. Then (2.20) has a solution iff

$$-acx^2 - bcy^2 - z^2 = 0 \tag{2.36}$$

has one (compare subsection 2.2.2). But (2.36) satisfies the requirements of Theorem 7. So we get

$$-ac\,R\,-bc, \qquad \text{(by (2.25))} \tag{2.37}$$

$$-bc\,R\,-ac, \qquad \text{(by (2.26))} \tag{2.38}$$

$$-\frac{(-ac)(-bc)}{c^2}\,R\,c. \quad \text{(by (2.27))} \tag{2.39}$$

Clearly (2.39) gives (2.21), while (2.37) implies (2.22) and (2.38) implies (2.23). ∎

Proof. (Theorem 7)
First of all we consider two special (simple) instances of (2.24)

(Case a = 1) Obviously, $(\overline{x}, \overline{y}, \overline{z}) = (1, 0, 1)$ is a solution, and (2.25) - (2.27) hold.

(Case a = b) (2.25) and (2.26) always hold in this case while (2.27) requires -1 to be a square modulo b. If this is the case, then by Lemma 8 (given immediately after this proof) we can find integers r and s such that $b = r^2 + s^2$, leading to a solution $(\overline{x}, \overline{y}, \overline{z}) = (r, s, r^2 + s^2)$. On the other hand, if $b(x^2 + y^2) = z^2$ has a nontrivial solution, so has $(x^2 + y^2) = bz^2$ (compare subsection 2.2.2). Choosing such a solution $(\overline{x}, \overline{y}, \overline{z})$ gives

$$\overline{x}^2 + \overline{y}^2 \equiv_b 0. \tag{2.40}$$

Since $\gcd(\overline{x}, b) = 1$ (remember that we always require $\gcd(\overline{x}, \overline{y}, \overline{z}) = 1$), we can choose $\overline{x}'$ with $\overline{x}\overline{x}' \equiv_b 1$. Multiplying (2.40) by $(\overline{x}')^2$ gives

$$(\overline{y}\overline{x}')^2 \equiv_b -1,$$

i. e. $-1\,R\,b$.

Now we proceed to the general case. We may assume $a > b$, for if $b > a$ just interchange the roles of x and y. The strategy will be the following : We construct a new form

$Ax^2 + by^2 = z^2$ satisfying the same hypotheses as (2.24), $0 < A < a$, and having a nontrivial solution iff (2.24) does so (and a solution of (2.24) can be computed from a solution of the new form). After a finite number of steps, interchanging A and b in case A is less than b, we arrive at one of the cases $A = 1$ or $A = b$, each of which has been settled. Now for the details.

We will not reprove the necessity of (2.25) - (2.27) (see the proof of the necessity of (2.21) - (2.23) in subsection 2.3.1 and the proof of the equivalence of Theorem 6 and Theorem 7 given above). Therefore we will now assume that (2.25) - (2.27) hold.

By (2.26) there exist integers x and k such that

$$x^2 = b + ka. \tag{2.41}$$

Let $k = Am^2$, where A is the squarefree part of k. Also note that we can choose x such that $|x| \leq a/2$ by choosing the absolute least residue of x modulo a ("*symmetric representation* of the integers modulo a"). Let us now restate (2.41) as

$$x^2 = b + Am^2a. \tag{2.42}$$

First of all we show that $0 < A < a$. Since

$$0 \leq x^2 \overset{\text{by (2.42)}}{=} b + Am^2a \overset{\text{since } b<a}{<} a + Am^2a = a(1 + Am^2)$$

we have $0 < 1 + Am^2$, and hence $A \geq 0$. But if $A = 0$, then (2.42) gives $x^2 = b$, contradicting the fact that b is squarefree. So we established $A > 0$. On the other hand

$$Am^2a \overset{\text{by (2.42) \& } b>0}{<} x^2 \overset{|x|\leq a/2}{\leq} \frac{a^2}{4},$$

and so we have $A \leq Am^2 < a/4 (< a)$. So we consider now

$$Ax^2 + bY^2 = Z^2. \tag{2.43}$$

Clearly A, b are positive and squarefree integers. So we want to show

$$A\,R\,b, \tag{2.44}$$

$$b\,R\,A, \tag{2.45}$$

$$-\frac{Ab}{\gcd(A,b)^2}\,R\,\gcd(A,b). \tag{2.46}$$

In addition, we need that (2.24) has a nontrivial solution iff (2.43) has one, which will be shown constructively.

ad (2.44) With $g = \gcd(a,b)$, let $b_1 = b/g$, $a_1 = a/g$. We show $A\,R\,g$ and $A\,R\,b_1$. Then, by Lemma 9 (see below) we have $A\,R\,b_1g$, i. e. $A\,R\,b$. First of all, note that (2.42) may be written as

$$x^2 = b_1g + Am^2a_1g. \tag{2.47}$$

Since g is squarefree we have that g divides x. Setting $x_1 = \frac{x}{g}$ and cancelling gives

$$gx_1^2 = b_1 + Am^2a_1. \tag{2.48}$$

Thus $Am^2a_1 \equiv_g -b_1$, and hence

$$Am^2a_1^2 \equiv_g -a_1b_1. \tag{2.49}$$

Also note that $\gcd(m,g) = 1$, since a common factor would divide b_1 (by (2.48)) and hence $b = b_1g$ would not be squarefree. But also $\gcd(a_1,g) = 1$ since $a = a_1g$ is squarefree. Let m' and a_1' be the inverses of m respectively a_1 modulo g. By (2.27) (i. e. by $-a_1b_1\,R\,g$) we may choose y such that $y^2 \equiv_g -a_1b_1$. Now (2.49) becomes $A \equiv_g (m')^2(a_1')^2y^2$, i.e. $A\,R\,g$. So this part is done. It remains to show $A\,R\,b_1$.
By (2.47) we have

$$x^2 \equiv_{b_1} Am^2a. \tag{2.50}$$

By (2.25) (i. e. by $a\,R\,b$) we have $a\,R\,b_1$. Note also that $\gcd(a, b_1) = 1$ since a common factor would divide b_1 and g, contradicting the fact that $b = b_1 g$ is squarefree. Similarly, $\gcd(m, b_1) = 1$ (use(2.47)). Let a^* and m^* be the inverses of a respectively m modulo b_1. Let z be such that $z^2 \equiv_{b_1} a$ and let z^* be its inverse modulo b_1. Now (2.50) becomes

$$A \equiv_{b_1} x^2(m^*)^2 a^* \equiv_{b_1} x^2(m^*)^2(z^*)^2,$$

i. e. $A\,R\,b_1$.

ad **(2.45)** By (2.42), we have $b\,R\,A$ immediately.

ad **(2.46)** With $r = \gcd(A, b)$ let $A_1 = A/r$, $b_2 = b/r$. We have to show $-A_1 b_2\,R\,r$. From (2.42) we conclude

$$x^2 = b_2 r + A_1 r m^2 a.$$

Since r is squarefree we have r divides x. So

$$\begin{aligned} A_1 m^2 a &\equiv -b_2 \pmod{r}, \text{ or} \\ -A_1 b_2 m^2 a &\equiv b_2^2 \pmod{r}. \end{aligned} \tag{2.51}$$

Since $\gcd(a, r) = \gcd(m, r) = 1$, we may choose a^+ and m^+ as the inverses of a respectively m modulo r. Furthermore, from (2.25) (i. e. from $a\,R\,b$) we obtain $a\,R\,r$. Choose w such that $w^2 \equiv_r a$. Denote by w^+ the inverse of w modulo r. Then (2.51) becomes

$$-A_1 b_2 \equiv_r b_2^2(m^+)^2 a^+ \equiv_r b_2^2(m^+)^2(w^+)^2,$$

i. e. $-A_1 b_2\,R\,r$.

So we established (2.44) - (2.46) for (2.43). Assume now that (2.43) has a nontrivial solution $(\overline{X}, \overline{Y}, \overline{Z})$. Then

$$A\overline{X}^2 = \overline{Z}^2 - b\overline{Y}^2. \tag{2.52}$$

Multiplying this by (2.42) (i. e. by $Am^2a = x^2 - b$) gives

$$\begin{aligned} a(A\overline{X}m)^2 &= (\overline{Z}^2 - b\overline{Y}^2)(x^2 - b) = \\ &= (\overline{Z}x + b\overline{Y})^2 - b(x\overline{Y} + \overline{Z})^2. \end{aligned}$$

Thus a solution of (2.24) is

$$\overline{x} = A\overline{X}m,\ \overline{y} = x\overline{Y} + \overline{Z},\ \overline{z} = \overline{Z}x + b\overline{Y}.$$

Written in matrix-form we have

$$\begin{bmatrix} \overline{x} \\ \overline{y} \\ \overline{z} \end{bmatrix} = \begin{bmatrix} Am & 0 & 0 \\ 0 & x & 1 \\ 0 & b & x \end{bmatrix} \cdot \begin{bmatrix} \overline{X} \\ \overline{Y} \\ \overline{Z} \end{bmatrix}.$$

The matrix is invertible since its two blocks are : the second (2×2) block has determinant $x^2 - b \neq 0$ (since b is squarefree). The solution is nontrivial since we claim that $\overline{x} = Am\overline{X} \neq 0$. Suppose $Am = 0$. Then by (2.42) we have $x^2 = b$, contradicting the squarefreeness of b. Suppose $\overline{X} = 0$. Then by (2.52) we have $\overline{Z}^2 = b\overline{Y}^2$, contradicting the squarefreeness of b. ■

Now we give the two lemmas that we owe to the reader.

Lemma 8 *For $r > 0$, $-1\, R\, r$ implies that*

$$x^2 + y^2 = r \tag{2.53}$$

has a nontrivial integral solution.

Lemma 9 *For relatively prime integers n_1, n_2 we have*

$$a\,R\,n_1 \text{ and } a\,R\,n_2 \text{ implies } a\,R\,n_1 n_2.$$

Proof. (Lemma 8)
Since $-1\,R\,r$, we may choose $x_0 \in N_0$ and $k \in N$ such that $x_0^2 = kr - 1$, i. e.

$$x_0^2 + 1 = kr. \tag{2.54}$$

Setting $y_0 = 1$, we can say that the equation $x^2 + y^2 = kr$ has the integral solution (x_0, y_0). We are done if $k = 1$. So suppose $k > 1$. We use the descent method (a common tool in number theory). We will construct k' with $k' < k$ (even $k' \leq k/2$) and x_2, $y_2 \in N_0$ such that $x_2^2 + y_2^2 = k'r$. Proceeding inductively, we will finally arrive at a solution of (2.53).

Let us consider $x_1 = x_0 \bmod k$, and $y_1 = y_0 \bmod k$ in symmetric representation of the integers modulo k. Now we have for some integers c, d

$$x_1^2 + y_1^2 = (x_0 - ck)^2 + (y_0 - dk)^2 \equiv_k x_0^2 + y_0^2 \overset{\text{by (2.54)}}{\equiv_k} 0.$$

Hence, for some k' we have $x_1^2 + y_1^2 = k'k$. Since

$$x_1^2 + y_1^2 \leq (\frac{k}{2})^2 + (\frac{k}{2})^2 = \frac{1}{2}kk$$

we have $k' \leq \frac{k}{2}$. In addition we have

$$k'k^2r = (k'k)(kr) = (x_1^2 + y_1^2)(x_0^2 + y_0^2) = (x_0x_1 + y_0y_1)^2 + (x_0y_1 - x_1y_0)^2.$$

Dividing by k^2 gives

$$k'r = (\frac{x_0x_1 + y_0y_1}{k})^2 + (\frac{x_0y_1 - x_1y_0}{k})^2.$$

So if $x_2 = (x_0x_1 + y_0y_1)/k$, $y_2 = (x_0y_1 - x_1y_0)/k$ are integers, we have a solution of $x^2 + y^2 = k'r$. But the numerators of x_2 (respectively y_2) are multiples of k :

$$x_0x_1 + y_0y_1 = x_0(x_0 - ck) + y_0(y_0 - dk) \equiv_k x_0^2 + y_0^2 \equiv_k 0,$$

and

$$x_0y_1 - x_1y_0 = x_0(y_0 - dk) - y_0(x_0 - ck) \equiv_k 0. \blacksquare$$

Proof. (Lemma 9)

Since $a\,R\,n_1$ and $a\,R\,n_1$ we may choose integers x_1, x_2 such that

$$x_1^2 \equiv_{n_1} a,\ x_2^2 \equiv_{n_2} a. \tag{2.55}$$

Since $\gcd(n_1, n_2) = 1$ we can choose (by the Extended Euclidean Algorithm) integers l_1, l_2 such that

$$l_1n_1 - l_2n_2 = x_2 - x_1,$$

or, equivalently,

$$x_1 + l_1n_1 = x_2 + l_2n_2. \tag{2.56}$$

By (2.55) we have

$$(x_1 + l_1n_1)^2 \equiv_{n_1} a,\ (x_2 + l_2n_2)^2 \equiv_{n_2} a. \tag{2.57}$$

Let now $g = x_1 + l_1n_1$. Combining (2.56) and (2.57) we arrive at

$$g^2 \equiv_{n_1} a,\ g^2 \equiv_{n_2} a,$$

i. e. for some integers k_1, k_2 we have

$$g^2 = a + k_1n_1 = a + k_2n_2. \tag{2.58}$$

(2.58) implies $k_1 n_1 = k_2 n_2$, and hence n_1 divides $k_2 n_2$. Since n_1, n_2 are relatively prime, n_1 divides k_2. So, for some integer c we have

$$k_2 = cn_1. \tag{2.59}$$

So, by (2.58) and (2.59) we have

$$g^2 = a + cn_1 n_2,$$

i. e. we have

$$g^2 \equiv_{n_1 n_2} a. \blacksquare$$

Remark 3 *In order to arrive at a rational point on the conic, we need not just any nontrivial solution* $(\overline{x}, \overline{y}, \overline{z})$ *but one with* $\overline{z} \neq 0$. *In the proof of Theorem 7 an equation like*

$$x^2 - y^2 + cz^2 = 0$$

(note $a = 1$*) is equipped with the solution* $(1, 1, 0)$. *Indeed, the existence of a solution whose z-component is different from* 0 *is always guaranteed in such a case (see Theorem 19 in section 4.2), e. g.*

$$(\overline{x}, \overline{y}, \overline{z}) = (1 - c, -1 - c, 2).$$

2.3.3 An algorithm for solving the Legendre Equation

Clearly the constructive proof for the existence of a nontrivial integral solution of (2.20) in subsection 2.3.2 (under the conditions given there) leads to an recursive algorithm for computing such a solution. We start with the subproblem considered in Lemma 8, namely the computation of a solution of (2.53). We assume the procedures *msqrt* ("modular squareroot"), that has the following meaning : for integers a, b with $a\,R\,b$ we have

$$msqrt(a, b)^2 \equiv_b a.$$

Such a procedure exists for example in Maple™. We work in symmetric representation of the integers modulo any number.

PROC Circle($\downarrow r \uparrow x \uparrow y$)

IN :

$r \in N$ with $-1\, R\, r$.

OUT :

$x, y \in Z$ such that $x^2 + y^2 = r$.

LOCAL

$k, x_1, y_1, h \in Z$.

BEGIN

$x := msqrt(-1, r)$; $y := 1$; $k := (x^2 + y^2)/r$;

while $k > 1$ **do**

$x_1 := x \bmod k$; $y_1 := y \bmod k$;

$h := (xx_1 + yy_1)/k$; $y := (xy_1 - x_1 y)/k$;

$x := h$; $k := (x^2 + y^2)/r$

end while

END Circle

In the proof of Lemma 8 we saw that k drops by a factor of 2 (at least) after each new assignment to it. The starting value of k can be estimated : $x^2 + 1 = kr$, where $|x| \leq \frac{r}{2}$. So we have

$$k = \frac{1}{r}(x^2 + 1) \leq \frac{1}{r}\frac{r^2}{4} = \frac{r}{4} < r.$$

So the number of executions of the while-loop in *Circle* is bounded by $\log(r)$.

In subsection 2.3.2 we saw how to reduce (2.20) to (2.24). This transformation will be handled by the procedure *Legendre*. For solving the transformed equation (2.24), it will call *LegendreHelp*, the procedure that does the recursive computation of a nontrivial integral solution according to the proof of Theorem 7 in subsection 2.3.2. Clearly, *Legendre* transforms (2.20) and calls *LegendreHelp* only if a solution exists. So it has to check the conditions (2.21) - (2.23) of Theorem 6 in subsection 2.3.2. Therefore we assume the procedure L (*"Legendre symbol"*), which has the following meaning : For integers a, b we have

$$L(a,b) = 1 \text{ iff } a\,R\,b,$$
$$L(a,b) = -1 \text{ otherwise.}$$

Thus we may test (2.21) - (2.23) in the following way : The conditions are satisfied iff

$$L(-ab,c) + L(-ac,b) + L(-bc,a) = 3.$$

Also L can be found in Maple™. Finally we need a procedure *sqfrp* (*"squarefree part"*) for computing the squarefree part of an integer (compare subsection 2.2.4). Now we can give the pseudocode.

PROC Legendre($\downarrow a \downarrow b \downarrow c \uparrow solvable \uparrow x \uparrow y \uparrow z$)

IN :

$a,\ b,\ c \in Z$:

nonzero, squarefree, pairwise relatively prime, not all positive nor negative.

OUT :

solvable : *boolean*.

($solvable = true$) iff $ax^2 + by^2 + cz^2 = 0$ has nontrivial integral solutions.

$x,\ y,\ z \in Z$:

nontrivial integral solution of $ax^2 + by^2 + cz^2 = 0$ if $solvable = true$.

BEGIN

$solvable := L(-ab,c) + L(-ac,b) + L(-bc,a) == 3;$

if *not solvable* **then** return;

if $(c < 0$ and $min(a,b) > 0)$ or $(c > 0$ and $max(a,b) < 0)$ **then**

Call *LegendreHelp*($\downarrow -ac, \downarrow -bc, \uparrow x, \uparrow y, \uparrow z$);

$z := z/c$

elseif $(a < 0$ and $min(b,c) > 0)$ or $(a > 0$ and $max(b,c) < 0)$ **then**

Call $LegendreHelp(\downarrow -ba, \downarrow -ca, \uparrow y, \uparrow z, \uparrow x)$;

$x := x/a$

else

Call $LegendreHelp(\downarrow -ab, \downarrow -cb, \uparrow y, \uparrow z, \uparrow x)$;

$y := y/b$

end if

END Legendre

PROC LegendreHelp($\downarrow a \downarrow b \uparrow x \uparrow y \uparrow z$)

IN :

$a,\, b \in Z$:

positive, squarefree with $a\,R\,b$, $b\,R\,a$, $-ab/gcd(a,b)^2\,R\,gcd(a,b)$.

OUT :

$x,\, y,\, z \in Z$

such that $ax^2 + by^2 = z^2$.

LOCAL

$r,\, s,\, T,\, A,\, B,\, X,\, Y,\, Z,\, m \in Z$

BEGIN

if $a == 1$ **then**

$x := 1;\ y := 0;\ z := 1$

elseif $a == b$ **then**

Call $Circle(\downarrow b, \uparrow x, \uparrow y)$;

$z := x^2 + y^2$

elseif $a > b$ **then**

$s := msqrt(b, a)$;

$T := (s^2 - b)/a$;

$A := sqfrp(T);\ m := sqrt(T/A)$;

Call $LegendreHelp(\downarrow A, \downarrow b, \uparrow X, \uparrow Y, \uparrow Z)$;

$x := AXm$; $y := sY + Z$; $z := sZ + bY$

else

$s := msqrt(a, b)$;

$T := (s^2 - a)/b$;

$A := sqfrp(T)$; $m := sqrt(T/B)$;

Call $LegendreHelp(\downarrow B, \downarrow a, \uparrow Y, \uparrow X, \uparrow Z)$;

$y := BYm$; $x := sX + Z$; $z := sZ + aX$

end if

END LegendreHelp

Some words on the number of self-references in *LegendreHelp*. The worst thing that can happen is that we reduce both coefficients of

$$ax^2 + by^2 = z^2$$

to 1. The number of self-references of *LegendreHelp* needed to achieve this is bounded by $2\log_4(\max(a, b))$, since every time we reduce a coefficient, it is reduced by a factor of 4 at least (see subsection 2.3.2). In the situation $a = b$ we call *Circle* (and no more call to *LegendreHelp* is needed), which calls himself not more than $log(a)$ times (as we know already). So in all cases, the maximal number of any procedure calls is $O(log(max(a, b))$.

2.4 Real points on rational conics

Let us again consider the general conic equation

$$g(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{2.60}$$

where a, b, c, d, e, f satisfy the conditions given in subsection 2.2.1 or 2.2.2. Remember especially that we posed conditions on the coefficients (in the hyperbolic/elliptic case) that guarantee the existence of more than one real point on the conic (e. g. $(c \neq 0 \wedge D > 0) \Rightarrow M_1 \neq 0$, see subsection 2.2.2). We do so, because we want to talk here only about conics whose graph constitutes something like a real curve (so we avoid cases like $x^2 + y^2 = -1$, or $x^2 + y^2 = 0$ that would define a purely complex set respectively a set containing only one real point).[3]

This time we assume that no rational point lies on the conic. In this case we ask whether there is at least a real point on the conic, i. e. whether there exists $(\overline{x}, \overline{y}) \in R^2$ such that

$$g(\overline{x}, \overline{y}) = 0.$$

Under the above assumptions, such a point always exists. Since we saw in subsection 2.2.1 that on every parabola lies a rational point, we only have to consider the elliptic/hyperbolic case. In subsection 2.2.2 we transformed (2.60) to an equation of the form

$$x^2 + Ky^2 = L, \tag{2.61}$$

where K, L are rational numbers satisfying $\neg(K > 0 \wedge L < 0)$. A real solution of (2.61) is given by

$$\begin{aligned} (\overline{x}, \overline{y}) &= (\sqrt{L}, 0) \text{ if } L > 0, \\ (\overline{x}, \overline{y}) &= (0, \sqrt{\frac{L}{K}}) \text{ if } L < 0. \end{aligned}$$

By retransforming, we arrive at a real solution to (2.60).

[3]The question when a rational algebraic plane curve over Q is parametrizable over R is treated in section 3.3 ("Parametrizing over the reals") of [SENDRA, WINKLER 96]. We state here the main result.

Theorem 10 (Thm. 3.2 of SENDRA, WINKLER 96) *A rational algebraic plane curve over Q is parametrizable over R if and only if it is not birationally equivalent over R to the conic $x^2 + y^2 + z^2$.*

2.4.1 Algorithm for the real case

The procedure *Conic2* given in subsection 2.2.4 already includes the formulas given above: After the call of the procedure *Legendre* the case that no rational point lies on the conic is treated in the following lines

CALL $Legendre(\downarrow a_2, \downarrow b_2, \downarrow c_2, \uparrow ratpoint, \uparrow x, \uparrow y, \uparrow z)$;

if not *ratpoint* **then**

  if $L > 0$ **then**

    $x := sqrt(L);\; y := 0$

  else

    $x := 0;\; y := \sqrt{L/K}$

  end if

else

....

The procedure then correctly retransforms this real solution too.

2.5 Concluding Remarks

In the parabolic case, we got a rational solution in form of a formula depending only on the coefficients of the defining polynomial and making use only of the field operations $+, \cdot, ^{-1}$ (compare subsection 2.2.1). Hence the problem of finding a rational point on a parabola is solved in general, i. e. for every field. Concerning the hyperbolic/elliptic case, we note that the reduction of the general conic equation over some field F to a reduced equation of the form

$$X^2 + KY^2 = L, \tag{2.62}$$

where $K, L \in F$, can also be performed using only the basic field operations (compare subsection 2.2.2). Hence we turn to the solvability of (2.62) for selected fields, e. g. for the field of rational functions over Q in the next chapter.

Chapter 3

Conics over Q(t)

3.1 Analogies with the rational case

As pointed out in section 2.5, we only have to consider the reduced equation

$$X^2 + K(t)Y^2 = L(t), \tag{3.1}$$

where K, $L \in Q(t)$. Our goal is to find rational functions $X(t)$, $Y(t)$ satisfying (3.1). This solves the problem of finding rational functions satisfying the general conic equation with coefficients in $Q(t)$ completely (compare chapter 2). For solving (3.1), we try to exploit the method used for the rational case. In order to point out the analogy between these cases, we note that $Q(t)$ is the quotient field of $Q[t]$, a *Euclidean Domain*[1] (ED for short), like Q is the quotient field of Z (*the* standard example of an ED). This means that we can make use of modular arithmetic, as we did in the rational case. Also those details of the rational case depending on factorization can be adapted, since every ED is

---

[1] A *Eucledian domain* is an integral domain J together with a "degree" function $d : J^* \to N$ such that

1. $\forall p_1, p_2 \in J^* : d(p_1 p_2) \geq d(p_1)$.
2. $\forall p_1 \in J \, \forall p_2 \in J^* \, \exists q, r \in J :$
   $p_1 = p_2 q + r \wedge (r = 0 \vee d(r) < d(p_2))$.

In the case $J = Q[t]$, d is the usual degree function for polynomials.

a *Unique Factorization Domain*[2] (UFD).

Now let us have a look at the concrete steps performed in the (sub)sections of chapter 2. First of all, we note that we can perform the homogenization of (3.1) as in subsection 2.2.2, leading to an equation of the form

$$a(t)x^2 + b(t)y^2 + c(t)z^2 = 0, \tag{3.2}$$

where $a,\ b,\ c \in Q[t]^*$. Indeed, when we looked at (3.2) over the integers, we also had a sign condition ("a, b, and c do not all have the same sign") whose role can be characterized like this : if it does not hold, then (3.2) has only the trivial solution (at least if we restrict ourselves to real solutions). The right generalization of this condition at this stage would be

$$\begin{array}{l} \text{for every real } t_0 \text{ we have} \\ a(t_0),\ b(t_0),\ c(t_0) \\ \text{are not all positive,} \\ \text{nor all negative.} \end{array} \tag{3.3}$$

(Note that this condition would be quite nasty to check). (3.3) is necessary in the following sense.

Lemma 11 (Necessity of Sign Condition) *Suppose (3.3) does not hold. Then the only polynomial solution of (3.2) is* $(x(t), y(t), z(t)) \equiv (0,0,0)$.

Proof. (Lemma 11)
Let $t_0 \in R$ be such that w. l. o. g.

$$a(t_0),\ b(t_0),\ c(t_0) > 0.$$

---
[2]A *Unique Factorization Domain* is a ring R in which every nonzero element $a \neq \pm 1$ can be written as $\pm$ the product of primes in at most one way, unique up to the order of the factors.

Since polynomial functions are continuous, we might choose $\epsilon > 0$ such that

$$\text{for all } t \in [t_0 - \epsilon, t_0 + \epsilon] \text{ we have}$$
$$a(t),\ b(t),\ c(t) \text{ are positive.}$$

Now we see that a solution $(x(t), y(t), z(t))$ of (3.2) has to satisfy

$$x(t) = y(t) = z(t) = 0 \text{ for all } t \in [t_0 - \epsilon, t_0 + \epsilon].$$

But polynomials vanishing on infinitely many points vanish everywhere, i. e.

$$x(t) \equiv y(t) \equiv z(t) \equiv 0. \blacksquare$$

Concerning the condition (3.3), we use a simple strategy : we ignore it. Indeed, at a much later stage, we can easily check whether the order of Q affects the solvability of (3.2) or not. If one wants to have a (sufficient) criterion that makes it possible to detect non-solvability at this early stage, then one could test whether $lc(a)$, $lc(b)$, and $lc(c)$ do all have the same sign; if so then

$$\lim_{t\to\infty} a(t) = \lim_{t\to\infty} b(t) = \lim_{t\to\infty} c(t) = \pm\infty.$$

Hence there exists t_0 such that $a(t_0)$, $b(t_0)$, and $c(t_0)$ do all have the same (nonzero) sign, and so by lemma 11 equation (3.2) has only the trivial solution. But clearly, this test is weaker than (3.3).

We are also able to perform the next step of subsection 2.2.2, namely to make a, b, and c squarefree and (pairwise) relatively prime. The latter is clear since we can compute the *greatest common divisor* of polynomials (by the *Euclidean algorithm* - this holds for every ED). But also a squarefree factorization of a polynomial can easily be done (and corresponding commands belong to the kernel of most Computer-Algebra systems).

Hence we arrive at an equation of the form

$$a(t)x^2 + b(t)y^2 + c(t)z^2 = 0, \tag{3.4}$$

where a, b, $c \in Q[t]^*$ are squarefree and pairwise relatively prime. Now the three conditions given in Legendre's Theorem Version 1 (Theorem 7, see subsection 2.3.2) for the rational case

$$-ab\,R\,c, \tag{3.5}$$

$$-ac\,R\,b, \tag{3.6}$$

$$-bc\,R\,a, \tag{3.7}$$

are also necessary in order that (3.4) is solvable. From now on we assume that we can decide for two polynomials $p_1(t)$, $p_2(t)$ whether p_1 is a quadratical residue modulo p_2 (written $p_1\,R\,p_2$), i. e. whether there exists a polynomial $q(t)$ such that

$$q(t)^2 \equiv p_1(t) \mod p_2(t).$$

So we assume the existence of a function *pmsqrt* with the property

$$p_1\,R\,p_2 \text{ implies } pmsqrt(p_1, p_2)^2 \equiv p_1 \mod p_2.$$

We will treat a method for computing such a polynomial-modular squareroot at the end of this section. In addition, we assume the procedure *sqfrp* ("squarefree part") and *psqrt* ("polynomial squareroot") that deliver the squarefree part respectively the squareroot of a polynomial (the latter only if the polynomial is a square).

After having verified that (3.5) - (3.5) hold, we can continue the reduction of (3.4) to

$$a(t)x^2 + b(t)y^2 = z^2 \tag{3.8}$$

as in subsection 2.3.2. Hence a and b are nonzero and squarefree polynomials satisfying

$$a\,R\,b, \tag{3.9}$$

$$b\,R\,a, \tag{3.10}$$

$$-\frac{ab}{\gcd(a,b)^2}\,R\,\gcd(a,b). \tag{3.11}$$

W. l. o. g. let us assume $\deg(a) \geq \deg(b)$. From the proof of Legendre's Theorem Version 2 (Theorem 7 in subsection 2.3.2) we know that in the new coordinates

$$\begin{aligned} x &= AXm, \\ y &= sY + Z, \\ z &= sZ + bY, \end{aligned}$$

where

$$\begin{aligned} s(t) &= pmsqrt(b(t), a(t)), \\ k(t) &= \frac{s(t)^2 - b(t)}{a(t)}, \\ A(t) &= sqfrp(k(t)), \\ m(t) &= psqrt(\frac{k(t)}{A(t)}), \end{aligned}$$

(3.8) has the form

$$AX^2 + bY^2 = Z^2.$$

In analogy to the rational case A is smaller than a in some sense : in subsection 2.3.3 it was the absolute value of a that dropped; here it is the degree of the polynomial $a(t)$ that drops.

Lemma 12 *Let* $a(t),\ b(t) \in Q[t]^*$ *with* $\deg(a) \geq \deg(b)$ *and* $\deg(a) \geq 2$ *such that* $b\,R\,a$.

Then for

$$k(t) = \frac{s(t)^2 - b(t)}{a(t)}, \quad \textit{where } s(t) = pmsqrt(b, a)$$

we have for some positive integer l

$$\begin{aligned} \deg(k) &= \deg(a) - 2l, \textit{ or} \\ \deg(k) &= 0. \end{aligned}$$

Proof. (Lemma 12)

First of all we note that $s(t)$ can be chosen such that $\deg(s) \leq \deg(a) - 1$. Let $l \in N$ be such that

$$\deg(s) = \deg(a) - l. \tag{3.12}$$

Suppose $\deg(s^2) < \deg(b)$. Since

$$s^2(t) = b(t) + k(t)a(t),$$

we get

$$\begin{aligned} \deg(s^2 - b) &= \deg(k) + \deg(a), \text{ i. e.} \\ \deg(k) &= \deg(b) - \deg(a) \leq 0, \end{aligned}$$

and hence $\deg(k) = 0$ (proving the second case of the lemma). So let us now assume that

$$\deg(s^2) \geq \deg(b). \tag{3.13}$$

Now we have

$$\begin{aligned} \deg(k) &= \deg(\frac{s^2 - b}{a}) = \deg(s^2 - b) - \deg(a) \overset{\text{by (3.13)}}{=} \\ &= \deg(s^2) - \deg(a) = 2\deg(s) - \deg(a) \overset{\text{by (3.12)}}{=} \end{aligned}$$

$$= \quad 2(\deg(a) - l) - \deg(a) = \deg(a) - 2l. \blacksquare$$

Since $A(t) = k(t)/m(t)^2$ we get

$$\begin{aligned}
\deg(A) &= \deg(k) - \deg(m^2) \overset{\text{by Lemma 12}}{=} \deg(a) - 2l - 2\deg(m) = \\
&= \deg(a) - 2n, \ \text{ where } n = l - \deg(m).
\end{aligned}$$

Hence the degree of $A(t)$ is smaller than the degree of $a(t)$ by a multiple of 2 (we skipped here the case $\deg(k) = 0$ which leads to $\deg(A) = 0$, an ideal situation).

Now, by iterated coordinate transformations (as long as the degree of either a or b is greater than 2), we will finally arrive at one of the following situations :

$$\deg(a) = \deg(b) = 1, \tag{3.14}$$

$$\deg(a) = 1, \ \deg(b) = 0 \text{ (or vice versa)}, \tag{3.15}$$

$$\deg(a) = \deg(b) = 0. \tag{3.16}$$

We will now treat these special cases.

ad (3.14) Since $\deg(s) \leq \deg(a) - 1$ we have $\deg(s) = 0$ and hence

$$\deg(k) = \deg(\frac{s^2 - b}{a}) = \deg(b) - \deg(a) = 0.$$

This implies $deg(A) = 0$ and we arrive at (3.15).

ad (3.15) Again $\deg(s) = 0$, i. e. we have

$$s^2 - b = k(t) \underbrace{(a_0 + a_1 t)}_{a(t)}.$$

By comparing degrees on both sides we get $k(t) \equiv 0$, i. e. $s^2 = b$. Hence $a(t)x^2 + b(t)y^2 = z^2$ has the solution $(0, 1, s)$.

ad (3.16) In this case we are confronted with an equation of the form

$$ax^2 + by^2 = z^2, \tag{3.17}$$

where a, $b \in Q$. Clearly, this case can be treated with the methods of chapter 2.

Remark 4 *It is sufficient to search an integral (rational) solution for equation (3.17), since any polynomial solution implies the existence of many rational solutions by "plugging in" (note that this argument works only because a and b do not depend on t !). Also the question of solvability is only decided at this stage : (3.17) might not have an integral solution (remember our discussion on the sign-condition for (3.2)). If (3.17) has a non-trivial integral solution, then we invert all coordinate transformations (as in the rational case), leading to a polynomial solution of (3.2) and finally to a rational function solution for (3.1) and the general conic equation.*

Now we turn to the problem of calculating (at least in principle) the squareroot of a polynomial modulo another polynomial.

3.1.1 Quadratical residues in $Q[t]$

Suppose we want to determine for two polynomials p_1, p_2 whether

$$p_1(t)\, R\, p_2(t).$$

We may assume $\deg(p_1) < \deg(p_2)$, otherwise we reduce p_1 modulo p_2. We make an ansatz $q(t)$ for the polynomial squareroot of p_1 modulo p_2 of degree $\deg(p_2) - 1$. The polynomial $q(t)$ has to satisfy

$$q(t)^2 \equiv p_1(t) \mod p_2(t), \text{ i. e.}$$

$$rem(q(t)^2 - p_1(t), p_2(t)) = 0.$$

This condition gives us equations for determining the unknown coefficients of $q(t)$. The question whether there are at all any rational solutions for these coefficients decides the question whether $p_1(t)\,R\,p_2(t)$ over $Q[t]$. Let us look at an example.

Example 4 *We want to decide whether*

$$t+1\,R\,t^2$$

holds, and if it does compute a squareroot of $t+1$ *modulo* t^2. *We make an ansatz of degree* $\deg(t^2)-1$ $(=1)$ *:*

$$q(t) = q_0 + q_1 t.$$

Now we have

$$q(t)^2 - (t+1) = q_1^2 t^2 + (2q_0q_1 - 1)t + (q_0^2 - 1).$$

Reducing this expression modulo t^2 *gives*

$$(2q_0q_1 - 1)t + (q_0^2 - 1).$$

Equating this remainder to 0 *leads to*

$$\begin{aligned} q_0^2 &= 1, \\ 2q_0q_1 &= 1. \end{aligned}$$

This system has the (rational) solutions $(q_0, q_1) = \pm(1, \frac{1}{2})$. *Hence we conclude* $t+1\,R\,t^2$ *and that* $q(t) = \pm(\frac{1}{2}t + 1)$ *is a polynomial squareroot of* $t+1$ *modulo* t^2. ∎

From this example we conclude that we deal in general with n polynomial equations (of degree 2) in n variables, where $n = \deg(p_2)$. We might use any of the known techniques to solve systems of polynomial equations (Gröbner bases, resultant computation, characteristic sets, ...). But indeed, this access was quite straight forward and its value lies more in demonstrating that we can (in principle) decide and compute the discussed items. A more practical access to this problem would be considering a squarefree factorization of p_1 modulo p_2.

3.2 Algorithms for $Q(t)$

In this section we give the modified algorithms for finding a rational function satisfying a general conic equation over $Q(t)$ (analogously to subsections 2.2.3, 2.2.4, and 2.4.1). First of all we deal again with the parabolic case. We assume the procedure *normalf*, delivering the *normal form* of a rational function.

PROC PARABOLA($\downarrow g \uparrow ok \uparrow \overline{x} \uparrow \overline{y}$)

IN:

$g \in Q(t)[x,y]$, $g(x,y) = ax^2 + bxy + cy^2 + dx + ey + f$.

OUT:

ok : boolean.

  ($ok = true$) iff $g(x,y) = 0$ defines an irreducible "parabola".

$\overline{x}, \overline{y} \in Q(t)$.

  ($ok = true$) implies $g(\overline{x}, \overline{y}) \equiv 0$.

LOCAL:

$d', f' \in Q(t)$.

BEGIN

$ok := ((a,d) \neq (0,0)) \wedge ((c,e) \neq (0,0)) \wedge (b^2 = 4ac) \wedge ((a,c) \neq (0,0))$;

if not ok **then** return;

if $(a \neq 0)$ **then**

  $(d', f') := (normalf(4ae - 2bd), normalf(4af - d^2))$;

  if $(d' \neq 0)$ **then**

    $\overline{y} := normalf(-f'/d'); \overline{x} := normalf(-(d + b\overline{y})/2a)$

  else

$ok := false$

end if

else # $(c \neq 0)$

$(d', f') := (normalf(4cd - 2be), normalf(4cf - e^2))$;

if $(d' \neq 0)$ **then**

$\overline{x} := normalf(-f'/d'); \overline{y} := normalf(-(e + b\overline{x})/2c)$

else

$ok := false$

end if

end if

END PARABOLA.

Now we turn to the analogous algorithm for the procedure *conic2* of subsection 2.2.4. We assume the procedures *numer* and *denom*, which deliver *numerator* and *denominator* of a rational function. In addition, $sqfrp$ should deliver the *squarefree part* of a polynomial, i.e. for $p = \prod_{i=1}^{r} p_i^i$, where the p_i are relatively prime, we have

$$sqfrp(p) = \prod_{i=1}^{r} p_i^{\mathrm{mod}(i,2)}.$$

The procedure *psqrt* should deliver the *polynomial squareroot* of a polynomial that represents a full square, i. e.

$$p(t) = q(t)^2 \Rightarrow psqrt(p) = q.$$

The procedure gcd should deliver the *greatest common divisor* of two (or more) polynomials. The procedure $lcoeff$ should deliver the *leading coefficient* of a polynomial, while *signum* delivers the *signum* of a rational number. Furthermore, we assume the procedure *Legendre* (given below) that decides whether the Legendre Equation has (nontrivial) polynomial solutions and eventually computes one (with $z \neq 0$). Operations like $+, -, \cdot, /$ are to be carried out in the field of the (nonzero) rational functions.

PROC CONIC2($\downarrow a \downarrow b \downarrow c \downarrow d \downarrow e \downarrow f \uparrow ok \uparrow ratpoint \uparrow X \uparrow Y$)

IN:

a, b, c, d, e, $f \in Q(t)$ defining the conic.

OUT:

ok, $ratpoint$: boolean.

$X, Y \in R(t)$.

($ok = true$) iff the general conic equation defines an irreducible "ellipse" or "hyperbola".

($ok = ratpoint = true$) implies that (X, Y) are rational functions over Q on the conic.

($ok = true$; $ratpoint = false$) implies that there is no rational function on the conic and that $(X, Y) = (fail, fail)$.

LOCAL

$D, K, L \in Q(t)$;

$x, y, z \in R(t)$;

$k_1, k_2, l_1, l_2, g, a_1, a_2, b_1, b_2, c_1, c_2, r_1, r_2, r_3, g_1, g_2, g_3 \in Q[t]$.

BEGIN

$D := normalf(4ac - b^2)$; $ok := D \neq 0$;

if not ok **then** return;

if $a = 0$ and $c = 0$ **then**

  $K := -1$; $L := normalf(4(de - bf))$

elseif $c \neq 0$ **then**

  $K := D$; $L := normalf(4c^2d^2 - 4bcde + 4ace^2 + 4b^2cf - 16ac^2f)$

else # $(a \neq 0 \wedge c = 0)$

  $K := D$; $L := normalf(4a^2e^2 - 4bade + 4b^2af)$

end if

$ok := L \neq 0$;

if not ok **then** return;

$k_1 := numer(K);\ k_2 := denom(K);$

$l_1 := numer(L);\ l_2 := denom(L);$

$g := \gcd(k_2, l_2);$

$a_1 := normalf(l_2 k_2/g);\ b_1 := normalf(k_1 l_2/g);\ c_1 := normalf(-l_1 k_2/g);$

$ok := \neg(signum(lcoeff(a)) = signum(lcoeff(b)) = signum(lcoeff(c)));$

if not ok **then** return;

$a_2 := sqfrp(a_1);\ r_1 := psqrt(normalf(a_1/a_2));$

$b_2 := sqfrp(b);\ r_2 := psqrt(normalf(b_1/b_2));$

$c_2 := sqfrp(c);\ r_3 := psqrt(normalf(c_1/c_2));$

$g := \gcd(a_2, b_2, c_2);$

$a_2 := normalf(a_2/g);\ b_2 := normalf(b_2/g);\ c_2 := normalf(c_2/g);$

$g_1 := \gcd(a_2, b_2);$

$a_2 := normalf(a_2/g_1);\ b_2 := normalf(b_2/g_1);\ c_2 := normalf(c_2 g_1);$

$g_2 := \gcd(a_2, c_2);$

$a_2 := normalf(a_2/g_2);\ b_2 := normalf(b_2 g_2);\ c_2 := normalf(c_2/g_2);$

$g_3 := \gcd(b_2, c_2);$

$a_2 := normalf(a_2 g_3);\ b_2 := normalf(b_2/g_3);\ c_2 := normalf(c_2/g_3);$

CALL $Legendre(\downarrow a_2, \downarrow b_2, \downarrow c_2, \uparrow ratpoint, \uparrow x, \uparrow y, \uparrow z);$

if $ratpoint$ **then**

  $x := normalf(x g_3/r_1);\ y := normalf(y g_2/r_2);\ z := normalf(z g_1/r_3);$

  $x := normalf(x/z);\ y := normalf(y/z)$

else

  if $(L \in Q) \wedge (K \in Q)$ **then**

    if $L > 0$ **then**

      $x := sqrt(L);\ y := 0$

    else

      $x := 0;\ y := \sqrt{L/K}$

end if

else

$X := fail$; $Y := fail$; **return**

end if

end if

if $a = 0$ and $c = 0$ **then**

$X := normalf((x + y - 2e)/2b)$; $Y := normalf((x - y - 2d)/2b)$

elseif $c \neq 0$ **then**

$X := normalf((x - 2dc + be)/K)$; $Y := normalf((y - bX - e)/2c)$

else # $(a \neq 0 \wedge c = 0)$

$Y := normalf((x - 2ea + bd)/K)$; $X := normalf((y - bY - d)/2a)$

end if

END CONIC2

Now we come to the generalizations of the procedures *Legendre* and *LegendreHelp* (compare subsection 2.3.3). We assume the procedure *iLSolve* that decides and solves the Legendre equation over the integers. Furthermore we need the boolean function *quadres* that decides whether a polynomial is a quadratical residue modulo another polynomial, i. e. for polynomials $a, b \in Q[t]$ we have

$$quadres(a, b) = true \text{ iff } a\,R\,b.$$

PROC Legendre($\downarrow a \downarrow b \downarrow c \uparrow solvable \uparrow x \uparrow y \uparrow z$)

IN :

a, b, $c \in Q[t]$:

nonzero, squarefree, pairwise relatively prime.

OUT :

solvable : *boolean.*

$(solvable = true)$ iff $a(t)x^2 + b(t)y^2 + c(t)z^2 = 0$ has nontrivial polynomial solutions.

x, y, $z \in Q[t]$:

nontrivial polynomial solution of $a(t)x^2 + b(t)y^2 + c(t)z^2 = 0$ if $solvable = true$.

BEGIN

if $a, b, c \in Q$ **then**

Call $iLSolve(\downarrow a, \downarrow b, \downarrow c, \uparrow solvable, \uparrow x, \uparrow y, \uparrow z)$;

Return

end if

$solvable := \ quadres(-ab, c) \wedge \ quadres(-ac, b) \wedge \ quadres(-bc, a)$;

if not $solvable$ **then return**;

Call $LegendreHelp(\downarrow -ba, \downarrow -ca, \uparrow solvable, \uparrow y, \uparrow z, \uparrow x)$;

$x := x/a$

END Legendre

For *LegendreHelp* we assume *pmsqrt*, a function that computes the squareroot of a polynomial modulo another polynomial, i. e. for $a, b \in Q[t]$ with $a\,R\,b$ we have

$$pmsqrt(a, b)^2 \equiv a \pmod{b}.$$

PROC LegendreHelp$(\downarrow a \downarrow b \uparrow solvable \uparrow x \uparrow y \uparrow z)$

IN :

a, $b \in Q[t]$:

squarefree polynomials with $a\,R\,b$, $b\,R\,a$, $-ab/gcd(a, b)^2\,R\,gcd(a, b)$.

OUT :

$solvable$: $boolean$.

$(solvable = true)$ iff there exist nonzero polynomials over Q such that $a(t)x^2 + b(t)y^2 = z^2$.

x, y, $z \in Q[t]$:

$(solvable = true)$ implies $a(t)x^2 + b(t)y^2 = z^2$.

LOCAL

r, s, T, A, B, X, Y, Z, $m \in Q[t]$.

BEGIN

if $degree(a) = 0$ **and** $degree(b) = 0$ **then**

Call $iLSolve(\downarrow a, \downarrow b, \downarrow -1, \uparrow solvable, \uparrow x, \uparrow y, \uparrow z)$

elseif $degree(a) = 0$ **and** $degree(b)$ odd **then**

$x := 1;$

$y := 0;$

$z := sqrt(a);$

$solvable := true$

elseif $degree(a) \geq degree(b)$ **then**

$s := pmsqrt(b, a);$

$T := normalf((s^2 - b)/a);$

$A := sqfrp(T); \; m := psqrt(normalf(T/A));$

Call $LegendreHelp(\downarrow A, \downarrow b, \uparrow solvable, \uparrow X, \uparrow Y, \uparrow Z);$

$x := normalf(AXm); \; y := normalf(sY + Z); \; z := normalf(sZ + bY)$

else

$s := pmsqrt(a, b);$

$T := normalf((s^2 - a)/b);$

$A := sqfrp(T); \; m := psqrt(normalf(T/B));$

Call $LegendreHelp(\downarrow B, \downarrow a, \uparrow solvable, \uparrow Y, \uparrow X, \uparrow Z);$

$y := normalf(BYm); \; x := normalf(sX + Z); \; z := normalf(sZ + aX)$

end if

END LegendreHelp

Some words on the number of self-references in *LegendreHelp*. The worst thing that can happen is that we reduce both coefficients of

$$a(t)x^2 + b(t)y^2 = z^2$$

to polynomials of degree 0. The number of self-references of *LegendreHelp* needed to achieve this is bounded by $\max(\deg(a), \deg(b))$, since every time we reduce the degree of either a or b, it is reduced by 2.

Chapter 4

Quadratic forms over arbitrary fields of characteristic $\neq 2$

In this chapter we describe some of the general properties of quadratic forms over arbitrary fields. We shall state some well-known results without proof. Throughout, K will denote an arbitrary field whose characteristic is not 2. The material is taken from [BOREVICH, SHAFAREVICH 66].

4.1 Equivalence of quadratic forms

By a quadratic form over the field K we mean a homogeneous polynomial of degree 2 with coefficients in K. Any quadratic form f can be written as (for some $n \in N$)

$$f = \sum_{i,j=1}^{n} a_{ij} x_i x_j,$$

where $a_{ij} = a_{ji} \in K$. The symmetric matrix

$$A = [a_{ij}]_{i,j=1,\dots,n}$$

is called the *matrix* of the quadratic form f. If the matrix is given, the quadratic form is completely determined (except for the names of the variables). The determinant $d = \det(A)$ is called the *determinant* of the quadratic form f. If $d = 0$ the form f is called *singular*, and otherwise it is called *nonsingular*. If we let X denote the column vector of the variables $x_1, x_2, ..., x_n$ (and so X^T is the row vector of the variables $x_1, x_2, ..., x_n$), then the quadratic form can be written as

$$f = X^T AX.$$

Suppose we replace the variables $x_1, x_2, ..., x_n$ by the new variables $y_1, y_2, ..., y_n$ according to the formula

$$x_i = \sum_{j=1}^{n} c_{ij} y_j \quad (1 \leq i \leq n, c_{ij} \in K).$$

In matrix form this linear substitution becomes

$$X = CY,$$

where Y is the column vector of the variables $y_1, y_2, ..., y_n$, and C is the matrix $[c_{ij}]_{i,j=1,...,n}$. If we replace the variables $x_1, x_2, ..., x_n$ in f by the corresponding expressions in $y_1, y_2, ..., y_n$, then (after carrying out the indicated operations) we shall obtain a quadratic form g (also over the field K) in the variables $y_1, y_2, ..., y_n$. The matrix A_1 of the quadratic form g equals

$$A_1 = C^T AC. \tag{4.1}$$

Two quadratic forms are called *equivalent*, and we write $f \sim g$, if there is a nonsingular change of variables which takes one form to the other. From formula (4.1) we obtain the following theorem.

Theorem 13 *If two quadratic forms are equivalent, then their determinants differ by a* *nonzero factor which is a square in K.*

Let γ be an element of K. If there exist elements $\alpha_1, \alpha_2, ..., \alpha_n$ in K for which

$$f(\alpha_1, \alpha_2, ..., \alpha_n) = \gamma,$$

then we say that the form f represents γ. In other words, a number is represented by a quadratic form if it is the value of the form for some values of the variables. It is easily seen that equivalent quadratic forms represent the same elements of the field K.

We shall further say that the form f represents zero in the field K if there exist values $\alpha_1, \alpha_2, ..., \alpha_n \in K$, *not all zero*, such that $f(\alpha_1, \alpha_2, ..., \alpha_n) = 0$. The property of representing zero is clearly preserved if we pass to an equivalent form.

Theorem 14 *If a quadratic form f in n variables represents an element $\alpha \neq 0$, then it is equivalent to a form of the type*

$$\alpha x_1^2 + g(x_2, ..., x_n),$$

where g is a quadratic form in $n-1$ variables.

Regarding the proof of this theorem we only note the following. If $f(\alpha_1, \alpha_2, ..., \alpha_n) = \alpha$, then not all α_i are equal to zero, so we can find a nonsingular matrix C, whose first row is $\alpha_1, \alpha_2, ..., \alpha_n$. If we apply to f the linear substitution whose matrix is C, we obtain a form in which the coefficient of the square of the first variable is α. The rest of the proof is carried out as usual.

If the matrix of a quadratic form is diagonal (that is, if the coefficient of every product of distinct variables equals zero), then we say that the form is *diagonal.* Theorem 14 now implies the following theorem.

Theorem 15 *Any quadratic form over K can be put in diagonal form by some nonsingular linear substitution. In other words, every form is equivalent to a diagonal form.*

In terms of matrices, Theorem 15 shows that for any symmetric matrix A there exists a nonsingular matrix C such that the matrix $C^T AC$ is diagonal.

4.2 Representation of field elements

Let n be a natural number.

Theorem 16 *If a nonsingular form represents zero in the field K, then it also represents all elements of K.*

Proof. Since equivalent forms represent the same field elements, it suffices to prove the theorem for a diagonal form $f = a_1x_1^2 + a_2x_2^2 + ... + a_nx_n^2$. Let $a_1\alpha_1^2 + a_2\alpha_2^2 + ... + a_n\alpha_n^2 = 0$ be a representation of zero, and let γ be any element of K. We can assume that $\alpha_1 \neq 0$. We express the variables $x_1, ..., x_n$ in terms of a new variable t :

$$x_1 = \alpha_1(1+t),\ x_k = \alpha_k(1-t) \quad (k = 2, ..., n).$$

Substituting in the form f we obtain

$$\begin{aligned} f^* &= f^*(t) = \\ a_1\alpha_1^2(1+t)^2 + \sum_{i=2}^{n} a_i\alpha_i^2(1-t)^2 &= \overbrace{\sum_{i=1}^{n} a_i\alpha_i^2}^{0} + t^2 \overbrace{\sum_{i=1}^{n} a_i\alpha_i^2}^{0} + 2a_1\alpha_1^2 t - \sum_{i=2}^{n} 2a_i\alpha_i^2 t = \\ 4a_1\alpha_1^2 t - \sum_{i=1}^{n} 2a_i\alpha_i^2 t &= 4a_1\alpha_1^2 t. \end{aligned}$$

If we now set $\bar{t} = \gamma/4a_1\alpha_1^2$, we obtain $f^*(\bar{t}) = \gamma$. ■

Theorem 17 *A nonsingular quadratic form f represents the element $\gamma \neq 0$ in K if and only if the form $-\gamma^2 x_0^2 + f$ represents zero.*

Proof. The necessity of the condition is clear. On the other hand assume that

$$-\gamma\alpha_0^2 + f(\alpha_1, ..., \alpha_n) = 0,$$

where not all α_i $(i \in \{0, 1, ..., n\})$ equal zero. If $\alpha_0 \neq 0$, then $\gamma = f(\alpha_1/\alpha_0, ..., \alpha_n/\alpha_0)$. If $\alpha_0 = 0$, then the form f represents zero, and hence by Theorem 16 it represents all elements of the field K. ■

Remark 5 *From the proof of Theorem 17 it is clear that if we determine all representations of zero by the form $-\gamma x_0^2 + f$ (only those in which $x_0 \neq 0$ are relevant), then we have also determined all representations of γ by the form f. Hence the question of the representability of an element of the field K by a nonsingular form can be reduced to the question of the representability of zero by a nonsingular form in one more variable.*

Theorem 18 *If a nonsingular form f represents zero, then it is equivalent to a form of the following type :*

$$y_1y_2 + g(y_3, .., y_n).$$

Proof. Using Theorem 16, we first find $\alpha_1, ..., \alpha_n$ such that $f(\alpha_1, ..., \alpha_n) = 1$. By Theorem 14 we can now put f in the form $x_1^2 + f_1(x_2, ..., x_n)$. Since the form $x_1^2 + f_1$ represents zero, we can find $\beta_2, ..., \beta_n$ such that $f_1(\beta_2, ..., \beta_n) = -1$. Again applying Theorem 14, we can put f_1 in the form $-x_2^2 + g(y_3, ..., y_n)$. Setting $x_1 - x_2 = y_1$, and $x_1 + x_2 = y_2$, we obtain the desired result. ■

Remark 6 *If we know some representation of zero by the form f, then all the operations described in the proof of Theorem 18 can be carried out explicitly, and the form $g(y_3, ..., y_n)$ can be determined. Now assume that for any quadratic form which represents zero over the field K, an actual representation of zero can be found. Then any nonsingular form can be transformed to a form of the type*

$$y_1y_2 + ... + y_{2s-1}y_{2s} + h(y_{2s+1}, ..., y_n), \tag{4.2}$$

where the form h does not represent zero. In any representation of zero by the form (4.2), at least one of the variables $y_1, y_2, ..., y_{2s-1}, y_{2s}$ must be nonzero. To determine all representations of zero in which, say, $y_1 = \alpha_1 \neq 0$, we note that we can give $y_3, ..., y_n$ arbitrary values $\alpha_3, ..., \alpha_n$ and then determine y_2 by the condition

$$\alpha_1 y_2 + \alpha_3\alpha_4 + ... + g(\alpha_{2s-1}, ..., \alpha_n) = 0.$$

This gives us an effective method for finding all representations of zero by a nonsingular quadratic form over the field K, ***provided that we have a method for determining whether or not a given form represents zero, and, in case it does, an algorithm for finding some specific representation of zero.***

Theorem 19 *Let the field K contain more than five elements. If the diagonal form*

$$a_1x_1^2 + ... + a_nx_n^2 \quad (a_i \in K)$$

represents zero in the field K, then there is a representation of zero in which all the variables take nonzero values.

Proof. We first show that if $a\zeta^2 = \lambda \neq 0$, then for any $b \neq 0$ there exist nonzero elements α and β such that $a\alpha^2 + b\beta^2 = \lambda$. To prove this fact we consider the identity

$$\frac{(t-1)^2}{(t+1)^2} + \frac{4t}{(t+1)^2} = 1.$$

Multiplying this identity by $a\zeta^2 = \lambda$, we obtain

$$a(\zeta\frac{t-1}{t+1})^2 + at(\frac{2\zeta}{t+1})^2 = \lambda. \tag{4.3}$$

Choose a nonzero γ in K so that the value of $t = t_0 = b\gamma^2/a$ is not ± 1. This can be done because each of the equations $bx^2 - a = 0$ and $bx^2 + a = 0$ has at most two solutions for

x in K, and the field K has more than five elements. Setting $t = t_0$ in (4.3), we obtain

$$a(\zeta\frac{t_0-1}{t_0+1})^2 + b(\frac{2\zeta\gamma}{t_0+1})^2 = \lambda,$$

and our assertion is proved. We can now easily complete the proof of the theorem. If the representation $a_1\zeta_1^2 + ... + a_n\zeta_n^2 = 0$ is such that $\zeta_1 \neq 0, ..., \zeta_r \neq 0, \zeta_{r+1} = ... = \zeta_n = 0$, where $r \geq 2$, then we have shown that we can find $\alpha \neq 0$ and $\beta \neq 0$ such that $a_r\zeta_r^2 = a_r\alpha^2 + a_{r+1}\beta^2$, and this yields a representation of zero in which the number of nonzero variables is increased by one. Repeating this process, we arrive at a representation in which all the variables have nonzero value. ■

4.3 Binary quadratic forms

A quadratic form in two variables is called *binary quadratic form.*

Theorem 20 *All nonsingular binary quadratic forms which represent zero in K are equivalent.*

Indeed, by Theorem 18, any such form is equivalent to the form y_1y_2.

Theorem 21 *In order that the binary quadratic form f with determinant $d \neq 0$ represents zero in K, it is necessary and sufficient that the element $-d$ be a square in K (that is, $-d = \alpha^2, \alpha \in K$).*

Proof. The necessity of the condition follows from Theorems 13 and 18. Conversely, if $f = ax^2 + by^2$ and $-d = -ab = \alpha^2$, then $f(\alpha, a) = a\alpha^2 + ba^2 = -ba^2 + ba^2 = 0$. ■

Theorem 22 *Let f and g be two nonsingular binary quadratic forms over the field K. In order that f and g be equivalent, it is necessary and sufficient that their determinants differ by a factor which is a square in K, and that there exists some nonzero element of K which is represented by both f and g.*

Proof. Both conditions are clearly necessary. To prove sufficiency, let $\alpha \neq 0$ be an element of K which is represented by both f and g. By Theorem 14 f and g are equivalent to the forms $f_1 = \alpha x^2 + \beta y^2$ and $g_1 = \alpha x^2 + \beta' y^2$. Since $\alpha\beta$ and $\alpha\beta'$ differ by a square factor, then $\beta' = \beta\gamma^2$, $\gamma \in K$, and this means that $f_1 \sim g_1$ and $f \sim g$. ■

Appendix A

Some numbertheoretic supplements

A.1 The Legendre Symbol

We give some facts (without proof) for the computation of the *Legendre Symbol* (and hence on the decision whether $a\,R\,b$ for integers a and b) using the *law of quadratic reciprocity*. We follow [SCHARLAU, OPOLKA 84].

Let p be an odd prime number and a an integer with $\gcd(p,a) = 1$. Legendre (Adrien-Marie, 1752 - 1833) defined the following symbol[1] :

$$\begin{aligned} L(a,p) &:= 1, \text{ if the congruence } x^2 \equiv_p a \text{ is solvable (i. e. if } a\,R\,p), \\ L(a,p) &:= -1, \text{ otherwise.} \end{aligned}$$

Today, $L(a,p)$ is called the *Legendre Symbol*. In the first case, a is called a quadratical residue modulo p, in the second, a quadratical nonresidue modulo p (compare the definition in section 2.3). The following theorem provides a first basis for computing the Legendre Symbol.

[1]Indeed, Legendre used the symbol

$$\left(\frac{a}{p}\right)$$

for this purpose, but we stay consistent with our notation from chapter 2.

Theorem 23 (Law of Quadratic Reciprocity) *Let p, q be prime numbers $\neq 2$. Then we have*

$$L(p,q)L(q,p) = (-1)^{\frac{1}{4}(p-1)(q-1)}, \tag{A.1}$$

and in addition

$$\begin{aligned} L(-1,p) &= 1, \text{ if } p \equiv_4 1, \\ L(-1,p) &= -1, \text{ if } p \equiv_4 3, \\ \text{i. e. } L(-1,p) &= (-1)^{\frac{1}{2}(p-1)}. \end{aligned} \tag{A.2}$$

Similarly

$$\begin{aligned} L(2,p) &= 1, \text{ if } p \equiv_8 1,7\ , \\ L(2,p) &= -1, \text{ if } p \equiv_8 3,5\ , \\ \text{i. e. } L(2,p) &= (-1)^{\frac{1}{8}(p^2-1)}. \end{aligned} \tag{A.3}$$

Formula (A.1) is called the Law of Quadratic Reciprocity. (A.2) is called the First Supplement to the Law of Quadratic Reciprocity. (A.3) is called the Second Supplement to the Law of Quadratic Reciprocity. ■

Remark 7 *(A.1) establishes a connection between $L(p,q)$ and $L/q,p)$. Offhand, it is not immediately clear that these two expressions are in any way related.*

Remark 8 *If p is an odd prime number, the multiplicative group F_p^* of the field F_p with p elements is cyclic of order 2. The kernel of the homomorphism $\lambda x.x^2$ ($\in Hom(F_p^*)$) has order 2. Therefore, $(F_p^*)^2$, the image of this homomorphism, has order $(p-1)/2$. This means that F_p^* contains the same number of squares as nonsquares : $[F_p^* : (F_p^*)^2] = 2$.*

(A.1) only deals with primes. We try to generalize the Legendre Symbol L here up to the point where its arguments can be two odd and relatively prime numbers : Let $\overline{a}$,

$\bar{b} \in F_p^*$ be two nonsquares. Then the product $\bar{a}\bar{b}$ is a square (compare remark 8). This leads to

$$L(ab, p) = L(a, p)L(b, p). \tag{A.4}$$

In addition, trivially

$$L(a, p) = L(a + kp, p) \tag{A.5}$$

for every integer k. For a "denominator" b that can be written as $b = p_1 p_2 ... p_k$ for some natural number k and prime numbers $p_1, p_2, ..., p_k$ we define

$$L(a, b) := L(a, p_1)L(a, p_2) \cdot ... \cdot L(a, p_k). \tag{A.6}$$

For odd a and b with $\gcd(a, b) = 1$ the following formula is a consequence of (A.1).

$$L(a, b)L(b, a) = (-1)^{\frac{1}{4}(a-1)(b-1)}. \tag{A.7}$$

Note that (A.7) may be written as

$$L(a, b) = L(b, a)(-1)^{\frac{1}{4}(a-1)(b-1)}, \tag{A.8}$$

since $L(b, a) \in \{-1, 1\}$ and hence $1/L(b, a) = L(b, a)$. Now we can easily compute the Legendre Symbol.

Example 5 *We want to compute $L(417, 383)$.*

$$\begin{aligned}
& L(417, 383) \overset{by\ (A.5)}{=} \\
= \ & L(34, 383) \overset{by\ (A.4)}{=} L(17, 383)L(2, 383) \overset{by\ (A.3)}{=} \\
= \ & L(17, 383) \cdot 1 \overset{by\ (A.8)}{=} L(383, 17)(-1)^{\frac{1}{4}(382 \cdot 16)} = \\
= \ & L(383, 17) \cdot 1 \overset{by\ (A.5)}{=} L(9, 17) \overset{by\ (A.8)}{=} L(17, 9)(-1)^{\frac{1}{4}(16 \cdot 8)} = \\
= \ & L(17, 9) \overset{by\ (A.5)}{=} L(7, 9) \overset{by\ (A.8)}{=} L(9, 7)(-1)^{\frac{1}{4}(8 \cdot 6)} = L(9, 7) \overset{by\ (A.5)}{=} \\
= \ & L(2, 7) \overset{by\ (A.3)}{=} 1. \ \blacksquare
\end{aligned}$$

A.2 A proof of the Legendre Theorem using Minkowski's Lattice Point Theorem

In this section we show how Legendre's Theorem follows of one *classical* result of number theory : *Minkowski's Lattice Point Theorem.* First of all we state it and give a (sketch of a) proof. We follow [SCHARLAU, OPOLKA 84].

Theorem 24 (Minkowski's Lattice Point Thm.) *Let L be a lattice*[2] *in R^n and K a centrally symmetric convex set around the origin, i. e., when x, $y \in K$, then $-x$ and $\frac{1}{2}(x+y) \in K$. Then, if*

$$vol(K) \geq 2^n \Delta(L),$$

the set K contains a lattice point $x \in L$, $x \neq 0$.

Proof. First of all let K be an arbitrary set with a well defined volume, such that K is disjoint from all the $K+x$, $x \in L^*$. Then we have $vol(K) \leq vol(E)$, where E is a fundamental domain. Intuitively, this is obvious; one proves it by decomposing K in pieces $K_1, K_2, ...$, where the pieces lie in different translates of the fundamental domain. Then one moves the pieces into a fixed fundamental domain where they are disjoint. This immediately gives our inequality (make a sketch). If $vol(K) > 2^n\Delta$, i. e., $vol(\frac{1}{2}K) > \Delta$ with $\frac{1}{2}K = \{\frac{1}{2}x | x \in K\}$, then not all the parallel translates of $\frac{1}{2}K$ are disjoint. Therefore there are $\frac{1}{2}x, \frac{1}{2}y \in \frac{1}{2}K$ and $z \in L$, $z \neq 0$ with $\frac{1}{2}x = \frac{1}{2}y + z$ or $z = \frac{1}{2}(x-y)$. By our assumption, $-y$ and $\frac{1}{2}(x-y)$ lie in K, which completes the proof. ∎

Now let us restate Legendre's Theorem.

---

[2]*Let $b_1, ..., b_n$ be linear independent column vectors of R^n $(n \geq 2)$. Then we call the set*

$$L = \{\sum_{k=1}^{n} \alpha_k b_k : \alpha_1, ..., \alpha_n \in Z\}$$

a lattice *in R^n. The number $\Delta(L) := |\det(b_1, ..., b_n)|$ equals the volume of the cuboid spanned by $b_1, ..., b_n$ (the so called* fundamental domain*) and is called the* volume of the lattice.

Theorem 25 (Legendre) *Let a, b, c be relatively prime square-free integers which do not all have the same sign. The equation*

$$ax^2 + by^2 + cz^2 = 0 \tag{A.9}$$

has a solution $(x, y, z) \neq (0, 0, 0)$ if and only if the following congruences are solvable :

$$\begin{aligned} u^2 &\equiv -bc \pmod{a}, && \text{(A.10)} \\ v^2 &\equiv -ca \pmod{b}, && \text{(A.11)} \\ w^2 &\equiv -ab \pmod{c}. && \text{(A.12)} \end{aligned}$$

A proof based on Minkowski's Lattice Point Theorem might run like this.

Proof. The necessity of (A.10) - (A.12) might be proved as usual (For $ax^2+by^2+cz^2 = 0$, one has $by^2 + cz^2 \equiv 0 \pmod{a}$ and consequently $(cz)^2 \equiv -bcy^2 \pmod{a}$. Since we can assume that x, y, z are relatively prime, y is a unit mod a, consequently $x^2 \equiv -bc \pmod{a}$ is solvable). Conversely, we consider the lattice L of all integral (x, y, z) with

$$\begin{aligned} uy &\equiv cz \pmod{a}, \\ vz &\equiv ax \pmod{b}, \\ wx &\equiv by \pmod{c} \end{aligned}$$

for a fixed solution (u, v, w) of the congruences (A.10) - (A.12). It is easy to see that $\Delta(L) = |abc|$ and that these congruences lead to the congruence

$$ax^2 + by^2 + cz^2 \equiv 0 \pmod{abc}, \; (x, y, z) \in L.$$

We know that the convex centrally symmetrical ellipsoid

$$K = \{(x, y, z) \in R^3 |\ |a|x^2 + |b|y^2 + |c|z^2 \leq R\}$$

has volume $4\pi/3\sqrt{R^3/|abc|}$. According to the Lattice Point Theorem, an element $(x, y, z) \in (L \cap K)$ with $(x, y, z) \neq (0, 0, 0)$ exists if

$$\frac{4\pi}{3}(\frac{R^3}{|abc|}) > 2^3|abc|$$

or

$$R > (\frac{6}{\pi})^{2/3}|abc|.$$

This means that $(x, y, z) \in L$ with $(x, y, z) \neq 0$ exists with

$$|ax^2 + by^2 + cz^2| \leq |a|x^2 + |b|y^2 + |c|z^2 < 2|abc|,$$

i. e., $ax^2 + by^2 + cz^2 = 0$ or $ax^2 + by^2 + cz^2 = \pm abc$. In the first case, we are finished. If $ax_0^2 + by_0^2 + cz_0^2 = -abc$, then

$$a(x_0 z_0 + by_0)^2 + b(y_0 z_0 - ax_0)^2 + c(z_0^2 + ab)^2 = 0,$$

and we are finished. To exclude the case $ax^2 + by^2 + cz^2 = abc$ the reader might use the fact that a, b, c do not all have the same sign. ■

Appendix B

Implementation and Examples

In this appendix I describe the Maple™-implemented programs for the user, and we will see some examples of finding rational points on conics using this implementation. A section containing the Maple™-code concludes the paper.

B.1 Information for the user

In this section we give an overview of the existing procedures and their function.[1]

The main procedure (and usually the only procedure a "normal" user will be confronted with) is the procedure *conic*. A usual call has the following form :

$$conic(poly, [x,y], 'ok', 'parabol', 'ratpoint', 'X', 'Y');$$

Here *poly* is the conic equation (a bivariate polynomial of degree 2) with coefficients in Q respectively $Q(t)$, i. e. *poly* is of the form $ax^2+bxy+cy^2+dx+ey+f$; its indeterminates are given by the list (here $[x,y]$) that forms the second argument of *conic*. *conic* has 5 output parameters whose role could be characterized as follows :

[1] Sometimes we loose some words on how a procedure works; this might not be understood by a reader not familiar with the theory of the previous chapters and hence he should skip such passages.

1. *ok* : we chose here this name for a boolean variable that checks whether the conditions we pose on the general conic equation are satisfied (irreducibility, conic not purely complex, ...). If so, then $ok = true$. Otherwise you may ignore the other output values.

2. *parabol* : the value of this boolean variable equals *true* iff *poly* defines a parabola.

3. *ratpoint* : the value of this boolean variable equals *true* iff there is a rational point on the conic.

4. (X, Y) : these two rational values satisfy $poly(X, Y) = 0$ (i. e. constitute a rational point on the conic) if $ratpoint = true$.

In addition to these output parameters, *conic* returns the value that results from substituting X and Y into *poly* as a function value (clearly only in case that there is a rational point on the conic). This value will always be 0, but this automatic verifying keeps one from being tempted to "plug in".

The tasks that *conic* performs are easily explained : it basically decides whether we deal with a parabola or an ellipse/hyperbola and deals with these cases by eventually calling the procedures *parabola* respectively *conic2*.

The procedure *parabola* receives as input values the six coefficients of the general conic equation and outputs a boolean variable indicating whether we really deal with a parabola, and if so, the coordinates of a rational point on the parabola. The rational point is computed by a formula that only depends on the coefficients of the defining polynomial. A typical call of this procedure looks like this :

$$parabola(a, b, c, d, e, f,' parabolic',' x',' y');$$

The procedure *conic2* receives as input values the six coefficients of the general conic equation and outputs two boolean values indicating whether we really deal with an ellipse/hyperbola and whether there is a rational point on this conic; furthermore it outputs two rational values representing the coordinates of a rational point on the conic (if one exists; otherwise those output parameters are set to $FAIL$). A typical call of this procedure looks like this :

$$conic2(a, b, c, d, e, f, 'ok', 'ratpoint', 'X', 'Y');$$

The major task of *conic2* is to transform the general conic equation to the corresponding Legendre equation, call *LegendreSolve* for finding an integral (respectively polynomial) solution of this Diophantine (respectively polynomial) equation, and finally to retransform it to a rational solution of the original conic equation.

The procedure *LegendreSolve* receives as input values the three coefficients of a Legendre equation and outputs a boolean value that indicates whether the equation is solvable. If it is, then the other three output variables contain an integral (respectively polynomial) solution of the Legendre equation. A typical call of this procedure looks like this :

$$LegendreSolve(a, b, c, 'solvable', 'x', 'y', 'z');$$

The procedure decides whether we solve a Diophantine or a polynomial equation and treats these two cases differently. In the first case the procedure uses *iLSolve* (a procedure based on Maples *isolve*) to solve the Diophantine equation. In the second case the procedure reduces the Legendre equation to one of a special form (the coefficient of the third variable is -1) and then uses *LegendreHelp* in order to get a polynomial solution of this equation. A retransformation of this solution leads to a polynomial solution of the original (more general) Legendre equation.

The procedure *LegendreHelp* receives as input values the two coefficients of a Legendre equation in special form (the coefficient of the third variable is -1) and outputs a boolean variable that indicates the solvability of the equation. If the equation is solvable, then the other three output variables contain a polynomial solution of the Legendre equation.

A typical call of this procedure looks like this :

$$LegendreHelp(a, b, 'solvable', 'x', 'y', 'z');$$

The task of *LegendreHelp* is to recursively transform the given Legendre equation to an equivalent one, thereby reducing the degrees of the coefficient polynomials. The basic case of an Legendre equation with rational coefficients is handled again by the procedure *iLSolve*.

Other procedures involved are *sqfrp*, a function that returns the squarefree part of a polynomial, and *quadres*, a procedure that decides whether a polynomial is a quadratical residue modulo another polynomial, and if so, computes the squareroot of this polynomial modulo the other.

B.2 Some examples

In this section we give some examples produced with a Maple^{TM}-implementation of the algorithms given in this paper (whose listing is shown in the next section). First of all one finds there the three examples carried out by hand in subsection 2.2.1. Then we give examples of conics that are either not irreducible, purely complex (i. e. do not contain a real point), or that only contain real points. An example for the non-parabolic case is taken from [SENDRA, WINKLER 96], section 3.2, namely the conic

$$x^2 - 4xy - 3y^2 + 4x + 8y - 5 = 0.$$

We try then to give a feeling for the growth[2] of the solutions by letting grow the constant in this polynomial, and finally we also change other coefficients in order to "blow up" the solution. Five examples with coefficients in $Q(t)$ conclude the section.

We use a short program (the procedure *ratpoint*) that calls the procedure *conic* and handles the output. Also the time needed for the calculation is given (in seconds). The examples were carried out on a 486 DX / 50 MHz (8 MB RAM).

[2] We understand groth here in the rational case in the sense of growing numerators and denominators.

```
> ratpoint := proc(g)
> local ok, parab, ratp, X, Y, result, timer;
> timer := time();
> result := conic(g,[x,y],'ok','parab','ratp','X','Y');
> timer := time() - timer;
> lprint(`Irreducible conic                     :`, ok);
> lprint(`Parabola                              :`, parab);
> lprint(`Existence of a rational point         :`, ratp);
> lprint(`Its x-coordinates                     :`); print(X);
> lprint(`Its y-coordinates                     :`); print(Y);
> lprint(`Conic equation evaluated there        :`, result);
> lprint(`Time needed for calculation           :`, timer);
> RETURN();
> end:
>
> g:=x^2+y: ratpoint(g);
  Irreducible conic               :  true
  Parabola                        :  true
  Existence of a rational point   :  true
  Its x-coordinates               :
```

$$0$$

```
  Its y-coordinates               :
```

$$0$$

```
  Conic equation evaluated there :  0
  Time needed for calculation    :  0

> g:=y^2+x+1: ratpoint(g);
  Irreducible conic               :  true
  Parabola                        :  true
  Existence of a rational point   :  true
  Its x-coordinates               :
```

$$-1$$

```
  Its y-coordinates               :
```

$$0$$

```
  Conic equation evaluated there :  0
  Time needed for calculation    :  0

> g:=x^2+2*x*y+y^2+x+2*y-2: ratpoint(g);
  Irreducible conic               :  true
  Parabola                        :  true
  Existence of a rational point   :  true
  Its x-coordinates               :
```

$$\frac{-11}{4}$$

```
  Its y-coordinates               :
```

$$\frac{9}{4}$$

```
  Conic equation evaluated there :  0
  Time needed for calculation    :  0
```

```
> g:= x^2 - 2*x*y + 3*y^2 - 2*x - 5*y + 3: ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$5$$

```
  Its y-coordinates                  :
```

$$3$$

```
  Conic equation evaluated there :   0
  Time needed for calculation     :   3.000

> g:= x^2 - 2*x*y + 3*y^2 - 2*x - 5*y + 3/2:  ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   false
  Its x-coordinates                  :
```

$$\frac{3}{4}\sqrt{15}+\frac{11}{4}$$

```
  Its y-coordinates                  :
```

$$\frac{1}{4}\sqrt{15}+\frac{7}{4}$$

```
  Conic equation evaluated there :   0
  Time needed for calculation     :   1.000

> g:= x^2 + y^2 - 1:  ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$0$$

```
  Its y-coordinates                  :
```

$$1$$

```
  Conic equation evaluated there :   0
  Time needed for calculation     :   0

> g:= x^2 + y^2 + 1:  ratpoint(g);
  Irreducible conic                  :   false
  Parabola                           :   false
  Existence of a rational point      :   fail
  Its x-coordinates                  :
```

$$\mathit{fail}$$

```
  Its y-coordinates                  :
```

$$\mathit{fail}$$

```
  Conic equation evaluated there :
  Time needed for calculation     :   0
```

```
> g := x^2 - y^2:  ratpoint(g);
  Irreducible conic                    :   false
  Parabola                             :   false
  Existence of a rational point        :   fail
  Its x-coordinates                    :
```

$$fail$$

```
  Its y-coordinates                    :
```

$$fail$$

```
  Conic equation evaluated there       :
  Time needed for calculation          :   0

> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 5: ratpoint(g);
  Irreducible conic                    :   true
  Parabola                             :   false
  Existence of a rational point        :   true
  Its x-coordinates                    :
```

$$0$$

```
  Its y-coordinates                    :
```

$$1$$

```
  Conic equation evaluated there       :   0
  Time needed for calculation          :   0

> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 5/4: ratpoint(g);
  Irreducible conic                    :   true
  Parabola                             :   false
  Existence of a rational point        :   true
  Its x-coordinates                    :
```

$$\frac{-1}{2}$$

```
  Its y-coordinates                    :
```

$$\frac{1}{3}$$

```
  Conic equation evaluated there       :   0
  Time needed for calculation          :   1.000

> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 5/57: ratpoint(g);
  Irreducible conic                    :   true
  Parabola                             :   false
  Existence of a rational point        :   true
  Its x-coordinates                    :
```

$$\frac{-29363}{10697}$$

```
  Its y-coordinates                    :
```

$$\frac{6160}{32091}$$

```
  Conic equation evaluated there       :   0
  Time needed for calculation          :   0
```

```
> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 53/57: ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$\frac{-829}{399}$$

```
  Its y-coordinates                  :
```

$$\frac{128}{399}$$

```
  Conic equation evaluated there :   0
  Time needed for calculation    :   1.000
```

```
> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 53/589: ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$\frac{97798934}{407851283}$$

```
  Its y-coordinates                  :
```

$$\frac{-50965683}{407851283}$$

```
  Conic equation evaluated there :   0
  Time needed for calculation    :   1.000
```

```
> g := x^2 - 4*x*y - 3*y^2 + 4*x + 8*y - 540/589: ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$\frac{74}{589}$$

```
  Its y-coordinates                  :
```

$$\frac{32}{589}$$

```
  Conic equation evaluated there :   0
  Time needed for calculation    :   0
```

```
> g := x^2 - 4*x*y - 321/412*y^2 + 4*x + 8*y - 540/589: ratpoint(g);
  Irreducible conic                  :   true
  Parabola                           :   false
  Existence of a rational point      :   true
  Its x-coordinates                  :
```

$$\frac{-3034547317966475 50}{5060385415656351}$$

```
  Its y-coordinates                  :
```

$$\frac{-197427631460316232}{15181156246969053}$$

```
  Conic equation evaluated there :   0
  Time needed for calculation    :   0
```

```
> g := x^2 - 4*x*y - 321/412*y^2 + 4*x + 842/523*y - 540/589: ratpoint(g);
  Irreducible conic                   :   true
  Parabola                            :   false
  Existence of a rational point       :   true
  Its x-coordinates                   :
```

$$-\frac{5278101767904785720565977157289 16}{48662725374633201583105054214834 27}$$

```
  Its y-coordinates                   :
```

$$\frac{617494015461956693895822488928 0132}{486627253746332015831050542148 3427}$$

```
  Conic equation evaluated there :    0
  Time needed for calculation    :    2.000
```

```
> g := x^2 - 4*x*y - 321/412*y^2 + 428/965*x + 842/523*y - 540/589; ratpoint(g);
```

$$g := x^2 - 4\,x\,y - \frac{321}{412}y^2 + \frac{428}{965}x + \frac{842}{523}y - \frac{540}{589}$$

```
  Irreducible conic                   :   true
  Parabola                            :   false
  Existence of a rational point       :   true
  Its x-coordinates                   :
```

$$-\frac{122614658596449155138741350495015311491277857212967386553312542}{408123878254880834898831398209104695060005236564334939877846 4795}$$

```
  Its y-coordinates                   :
```

$$\frac{111425209694055737082750301014886754162534735529967874784869970 32}{122437163476464250469649419462731408518001570969300481963353943 85}$$

```
  Conic equation evaluated there :    0
  Time needed for calculation    :    27.000
```

```
> g := 52/27*x^2+22/47*x*y-17/39*y^2-61/14*x+41/18*y-17/13; ratpoint(g);
```

$$g := \frac{52}{27}x^2 + \frac{22}{47}x\,y - \frac{17}{39}y^2 - \frac{61}{14}x + \frac{41}{18}y - \frac{17}{13}$$

```
  Irreducible conic                   :   true
  Parabola                            :   false
  Existence of a rational point       :   true
  Its x-coordinates                   :
```

$$-\frac{4526436000505375613648 19}{3523608727824695920668 167}$$

```
  Its y-coordinates                   :
```

$$\frac{1596868359964732251999447}{4607796028693833127027603}$$

```
  Conic equation evaluated there :    0
  Time needed for calculation    :    1.000
```

```
> g := -t/(1-t)*x^2 + 2*t*x*y - 2*t*y^2 - 1*x + 3*y + 2; ratpoint(g);
```

$$g := -\frac{t\,x^2}{1-t} + 2\,t\,x\,y - 2\,t\,y^2 - x + 3\,y + 2$$

```
Irreducible conic                 :   true
Parabola                          :   false
Existence of a rational point     :   true
Its x-coordinates                 :
```

$$-2\,\frac{-1+t}{t}$$

```
Its y-coordinates                 :
```

$$-\frac{1}{2}\,\frac{4\,t-3}{t}$$

```
Conic equation evaluated there  :   0
Time needed for calculation     :   21.000
```

```
> g := x^2 - 2*x*y + 4*t/(1-t)*y - 2; ratpoint(g);
```

$$g := x^2 - 2\,x\,y + 4\,\frac{t\,y}{1-t} - 2$$

```
Irreducible conic                 :   true
Parabola                          :   false
Existence of a rational point     :   true
Its x-coordinates                 :
```

$$-\frac{1}{2}\,\frac{-1+t^2+6\,t}{-1+t}$$

```
Its y-coordinates                 :
```

$$-\frac{1}{4}\,\frac{7+t^2+10\,t}{-1+t}$$

```
Conic equation evaluated there  :   0
Time needed for calculation     :   2.000
```

```
> g:= t*x^2 - 2*x*y + (3/t)*y - 2; ratpoint(g);
```

$$g := t\,x^2 - 2\,x\,y + 3\,\frac{y}{t} - 2$$

```
Irreducible conic                 :   true
Parabola                          :   false
Existence of a rational point     :   true
Its x-coordinates                 :
```

$$\frac{1}{4}\,\frac{-3+8\,t}{t}$$

```
Its y-coordinates                 :
```

$$-\frac{1}{8}+t$$

```
Conic equation evaluated there  :   0
Time needed for calculation     :   0
```

```
> g := x^2 - 2*t/(4-t)*x*y + z^2 - 5*t; ratpoint(g);
```

$$g := x^2 - 2\frac{t\,x\,y}{4-t} + z^2 - 5\,t$$

```
Irreducible conic                  :   true
Parabola                           :   false
Existence of a rational point      :   true
Its x-coordinates                  :
```

$$4$$

```
Its y-coordinates                  :
```

$$\frac{1}{8}\frac{\left(-16 - z^2 + 5\,t\right)(-4+t)}{t}$$

```
Conic equation evaluated there  :   0
Time needed for calculation     :   4.000
```

```
> g := t/(t^2+3) * x^2 - 2*y + 3; ratpoint(g);
```

$$g := \frac{t\,x^2}{t^2+3} - 2\,y + 3$$

```
Irreducible conic                  :   true
Parabola                           :   true
Existence of a rational point      :   true
Its x-coordinates                  :
```

$$0$$

```
Its y-coordinates                  :
```

$$\frac{3}{2}$$

```
Conic equation evaluated there  :   0
Time needed for calculation     :   0
```

B.3 The Maple^{TM} code

This section shows the listing of the implementation used in the previous section. One will find two versions. The first one can only handle the case where the coefficients of the conic equation are in Q. The second version treats the case $Q(t)$, but also handles the purely rational case correctly (it uses Maples procedure *isolve* to treat Diophantine equations). It is this version that produced the examples in the previous section and that is documented for the user. The first version can be regarded as straight-forward implementation of the algorithms in chapter 2, and as a basis for the second version.

```
> parabola := proc(a,b,c,d,e,f,ok,x,y)
>             local dp, fp, x1, y1;
> #
> #           IN :
> #           a,b,c,d,e,f : fraction.
> #
> #           OUT :
> #           ok : boolean.
> #                (ok = true)  means that
> #                a*x^2 + b*x*y + c*y^2 + d*x + e*y + f = 0          (GCE)
> #                defines an irreducible parabola.
> #           x, y : fraction.
> #                  (ok = true) implies that x and y satisfy (GCE).
> #
> #           LOCAL :
> #           dp, fp : fraction.
> #
>
>           ok := b^2 = 4*a*c and not (a=0 and c=0) and not (a=0 and d=0)
>                              and not (c=0 and e=0);
>           if not ok then RETURN() fi;
>
>           if f = 0 then x := 0; y := 0; RETURN() fi;
>                   if a <> 0 then
>                             dp := 4*a*e - 2*b*d;
>                             fp := 4*a*f - d^2;
>                             if dp <> 0 then
>                                y1 := - fp/dp;
>                                y := y1;
>                                x := - (d+b*y1)/(2*a)
>                             else
>                                ok := false
>                             fi
>                   else
>                             dp := 4*c*d - 2*b*e;
>                             fp := 4*c*f - e^2;
>                             if dp <> 0 then
>                                       x1 := - fp/dp;
>                                       x := x1;
>                                       y := - (e+b*x1)/(2*c)
>                             else
>                                       ok := false
>                             fi
>                   fi;
>
>                   RETURN()
>
> end:
>
```

```
> circle := proc(r,x,y)
>           local h, k, x1, y1, x2, y2;
> #
> #         IN :
> #         r : integer.
> #             r has to satisfy : r > 0,
> #             -1 is a quadratical residue modulo r (written -1 R r).
> #
> #         OUT :
> #         x, y : integer.
> #                They satisfy                  x^2 + y^2 = r.     (CE)
> #
> #         LOCAL :
> #         h, k, x1, y1, x2, y2 : integer.
> #
>           `mod` := mods;                      # symmetric representation
>           x1 := numtheory[imagunit] (r); # now  x1^2 = -1 mod r.
>           y1 := 1;
>           k := (x1^2 + y1^2) / r;
>
>           while k > 1 do
>                 x2 := x1 mod k;
>                 y2 := y1 mod k;
>                 h := (x1*x2 + y1*y2) / k;
>                 y1 := (x1*y2 - y1*x2) / k;
>                 x1 := h;
>                 k := (x1^2 + y1^2) / r
>           od;
>
>           x := x1;
>           y := y1;
>
>           RETURN()
>
> end:
>
>
```

```
> sqfrp := proc(n)
>           local factorlist, factornumber, signu, factorof, exponent, result, i;
> #
> #         IN :
> #         n : integer;
> #
> #         OUT :
> #         the squarefree part of n as function value.
> #
> #         LOCAL :
> #         factorlist : list.
> #         factornumber, factorof, signu, exponent, result, i : integer.
> #
>           if n = 0 then RETURN(1) fi;
>           readlib(isqrfree);
>           factorlist := isqrfree(n);
> #         Now factorlist = [ sign(n), [[f(1),e(1)],...,[f(m),e(m)]] ],
> #         where n = sign(n)*f(1)^e(1)*...*f(m)^e(m) is a squarefree
> #         factorization of n.
> #         The squarefree part of n can then be expressed as
> #         sign(n)*f(1)^(e(1) mod 2)*...*f(m)^(e(m) mod 2).
>
>           factornumber := nops(factorlist[2]);
>           result := 1;
>           signu := factorlist[1];
>
>           for i to factornumber do
>               factorof := factorlist[2][i][1];
>               exponent := factorlist[2][i][2];
>               result := result * factorof^(exponent mod 2)
>           od;
>
>           result := result * signu;
>
> end:
>
>
```

```
> LegendreHelp := proc(a,b,x,y,z)
>                 local r, s, T, A, B, m, X, Y, Z;
> #
> #               IN :
> #               a, b : integer.
> #                      a, b are positive and squarefree and satisfy
> #                      a R b, b R a, - a*b/gcd(a,b)^2 R gcd(a,b).
>
> #
> #               OUT :
> #               x, y, z : integer.
> #                         They satisfy  a*x^2 + b*y^2 = z^2.
> #
> #               LOCAL : r, s, T, A, B, m,  X, Y, Z : integer.
> #
>
>               `mod` := mods;         # symmetric representation
>
>               if a = 1 then
>                  x := 1;
>                  y := 0;
>                  z := 1
>               elif a = b then
>                    circle(b,'X','Y');
>                    x := X;
>                    y := Y;
>                    z := X^2 + Y^2
>               elif a > b then
>                    s := numtheory[msqrt] (b,a); # now s^2 = b mod a.
>                    T := (s^2 - b)/a;
>                    A := sqfrp(T);
>                    m := sqrt(T/A);
>                    LegendreHelp(A,b,'X','Y','Z');
>                    x := A*X*m;
>                    y := s*Y+Z;
>                    z := s*Z+b*Y
>               else
>                    s := numtheory[msqrt] (a,b); # now s^2 = a mod b.
>                    T := (s^2 - a)/b;
>                    B := sqfrp(T);
>                    m := sqrt(T/B);
>                    LegendreHelp(B,a,'Y','X','Z');
>                    y := B*Y*m;
>                    x := s*X+Z;
>                    z := s*Z+a*X
>               fi;
>
>               RETURN()
>
> end:
>
```

```
> LegendreSolve := proc(a,b,c,solvable,x,y,z)
>                  local X, Y, Z;
> #
> #                IN :
> #                a, b, c : integer.
> #                          a, b, c are nonzero, squarefree, pairwise
> #                          relatively prime and not all positive nor
> #                          all negative.
> #
> #                OUT :
> #                solvable : boolean.
> #                x, y, z : integer.
> #                          (solvable = true) implies that
> #                           a*x^2 + b*y^2 + c*z^2 = 0        (LE)
> #                          has a nontrivial integer solution.
> #                          If so, then (x,y,z) is one with z <> 0.
> #
> #                LOCAL :
> #                X, Y, Z : integer.
> #
>                  solvable :=
>                             evalb(numtheory[L](-a*b,abs(c)) +
>                                   numtheory[L](-a*c,abs(b)) +
>                                   numtheory[L](-b*c,abs(a)) =3);
>                              if not solvable then RETURN() fi;
> #
> #                           The first "if" avoids z = 0.
> #
>                  if abs(a) = 1 and abs(b) = 1 and a <> b then
>
>                     if abs(c) = 1 then
>                        if a = c then # Case +- (x^2 - y^2 + z^2) = 0.
>                                 x := 0; y := 1; z := 1
>                        else # Case +- (x^2 - y^2 - z^2) = 0.
>                                 x := 1; y := 0; z := 1
>                        fi
>                     else
>                         if a = 1 then # Case +- (x^2 - y^2 + c*z^2) = 0.
>                                 x := 1-c; y := -1-c; z := 2
>                         else # Case +- (-x^2 + y^2 + c*z^2) = 0.
>                                 x := -1-c; y := 1-c; z :=2
>                         fi
>                     fi;
>                     RETURN()
>                  fi;
>
>                  if (c < 0 and min(a,b) > 0) or (c > 0 and max(a,b) < 0) then
>                     LegendreHelp(-a*c,-b*c,'x','y','Z');
>                     z := Z/c
>                  elif (a < 0 and min(b,c) > 0) or (a > 0 and max(b,c) <0) then
>                     LegendreHelp(-a*b,-a*c,'y','z','X');
>                     x := X/a
>                  else
>                     LegendreHelp(-a*b,-b*c,'x','z','Y');
>                     y := Y/b
>                  fi;
>
>                  RETURN()
>
> end:
>
```

```
> conic2 := proc(a,b,c,d,e,f,ok,ratpoint,X,Y)
>           local D, K, L, k1, k2, l1, l2, g, a1, a2, b1, b2, c1, c2,
>                 r1, r2, r3, g1, g2, g3, x, y, y1, z, ratp, ok1;
> #
> #         IN :
> #         a, b, c, d, e, f : rational.
> #                             They define a conic via
> #            g(x,y) = a*x^2 + b*x*y + c*y^2 + d*x + e*y + f = 0.  (GCE)
> #
> #         OUT :
> #         ok, ratpoint : boolean.
> #         X, Y         : real.
> #                        (ok = true) implies that (GCE) defines an
> #                         irreducible ellipse or hyperbola with at least
> #                         two real points.
> #                        (ok = ratpoint = true) implies that (X,Y) are
> #                         rational coordinates of a point on the conic.
> #                        (ok = true; ratpoint = false) implies that (X,Y) are
> #                         real coordinates of a point on the conic and that
> #                         no rational point lies on the conic.
> #
> #         LOCAL :
> #         D, K, L : rational.
> #         x, y, z : real.
> #         k1, k2, l1, l2, g, a1, a2, b1, b2, c1, c2, r1, r2, r3,
> #         g1, g2, g3 : integer.
> #         ratp, ok1 : boolean.
> #
>         D := 4*a*c - b^2;
>         ok1 := evalb(D <> 0);
>         if not ok1 then ok := ok1; RETURN() fi;
> #
> #       Transformation of (GCE) in dependence of the values of a and c.
> #
>         if a = 0 and c = 0 then
>            K := -1;
>            L := 4*d*e - 4*b*f
>         elif c <> 0 then
>            K := D;
>            L := 4*c^2*d^2 - 4*b*c*d*e + 4*a*c*e^2 + 4*b^2*c*f - 16*a*c^2*f
>         else # if a<>0 and c=0
>            K := D;
>            L := 4*a^2*e^2 - 4*b*a*d*e + 4*b^2*a*f
>         fi;
> #
> #       Now (GCE) is equivalent to
> #                    x^2 + K*y^2 = L.                      (TE)
> #
```

```
>             ok1 := L <> 0 and not(K > 0 and L<0); # degenerate cases
>             ok := ok1;
>             if not ok1 then RETURN() fi;
> #           Some very special case :
>             if f = 0 then ratpoint := true; X := 0; Y := 0; RETURN() fi;
>
>             k1 := numer(K); k2 := denom(K);
>             l1 := numer(L); l2 := denom(L);
>             g := gcd(k2,l2);
>             a1 := l2*k2 / g; b1 := k1*l2 / g; c1 := - l1*k2 / g;
> #
> #           Now (TE) is equivalent to the diophantine equation
> #                   a1*x^2 + b1*y^2 + c1*z^2 = 0.              (DE)
> #
>             a2 := sqfrp(a1); r1 := sqrt(a1/a2);
>             b2 := sqfrp(b1); r2 := sqrt(b1/b2);
>             c2 := sqfrp(c1); r3 := sqrt(c1/c2);
>             g := gcd(a2,gcd(b2,c2));
>             a2 := a2/g; b2 := b2/g; c2 := c2/g;
>             g1 := gcd(a2,b2);
>             a2 := a2/g1; b2 := b2/g1; c2 := c2*g1;
>             g2 := gcd(a2,c2);
>             a2 := a2/g2; b2 := b2*g2; c2 := c2/g2;
>             g3 := gcd(b2,c2);
>             a2 := a2*g3; b2 := b2/g3; c2 :=c2/g3;
> #
> #           Here, (DE) is equivalent to
> #                     a2*(x')^2 + b2*(y')^2 + c2*(z')^2 = 0,
> #           where x' = x * r1 / g3, y' = y * r2 / g2, z' = z * r3 / g1.
> #           In addition, a2, b2, c2 satisfy the input-requirements
> #           of LegendreSolve.
> #
>             LegendreSolve(a2,b2,c2,'ratp','x','y','z');
>             ratpoint := ratp;
>             if ratp then # we arrive at a rational solution for (TE).
>                   x := x * g3 / r1; y := y * g2 / r2; z := z * g1 / r3;
>
>                   x := x / z; y := y / z
>             else # we only arrive at a real solution for (TE).
>                 if L > 0 then
>                         x := sqrt(L); y := 0
>                 else
>                         x := 0; y := sqrt(L/K)
>                 fi
>             fi;
> #
> #           Retransformation
> #
>             if a = 0 and c = 0 then
>                X := (x + y - 2*e) / (2*b); Y := (x - y - 2*d) / (2*b)
>             elif c <> 0 then
>                  x := (x - 2*d*c + b*e) / K; Y := (y - b*x - e) / (2*c);
>                  X := x
>             else
>                 y1 := (x - 2*e*a + b*d) / K; X := (y - b*y1 - d) / (2*a);
>                 Y := y1
>             fi;
>
>             RETURN()
>
> end:
>
```

```
> conic := proc(p,var,ok,parabol,ratpoint,X,Y)
>          local a, b, c, d, e, f, ok1, x1, y1, x, y;
> #
> #        IN :
> #        p : polynomial[var[1],var[2]].
> #            deg(p) := 2.
> #        var : list of the form [var1, var2], where var1, var2 are the 2
> #              undeterminates occuring in p.
> #
> #        OUT :
> #        ok, parabol, ratpoint : boolean.
> #        X, Y : real or "fail".
> #               (ok = true) implies that g(x,y) = 0 defines an irreducible
> #                conic with at least two real points.
> #               (ok = parabol = true) implies that g(x,y) = 0 defines a
> #                parabola.
> #               (ok = ratpoint = true) implies that (X,Y) are the coordinates
> #                of a rational point on the conic.
> #               (ok = true; ratpoint = false) implies that there is no
> #                rational point on the conic and that (X,Y) are coordinates
> #                of a real point on the conic.
> #        g(X,Y) as function value (verification : it has to be 0).
> #
> #        LOCAL :
> #        a, b, c, d, e, f : fraction.
> #        ok1 : boolean.
> #        x1, x2 : real.
> #        x, y : undeterminates.
> #
>          x := var[1]; y := var[2];
>          a := coeff(p,x^2);
>          b := coeff(coeff(p,x),y);
>          c := coeff(p,y^2);
>          d := coeff(coeff(p,x),y,0);
>          e := coeff(coeff(p,y),x,0);
>          f := coeff(coeff(p,x,0),y,0);
> #
> #        Now p = a*x^2 + b*x*y + c*z^2 + d*x + e*y + f.
> #
```

```
>           if b^2 = 4*a*c then # parabolic case
>              parabol := true;
>              parabola(a,b,c,d,e,f,'ok1','x1','y1');
>              ok := ok1;
>              if ok1 then
>                     ratpoint := true;
>                     X := x1; Y := y1
>              else
>                     ratpoint := fail; # interpret fail as "does not matter"
>                     X := fail; Y := fail
>              fi
>           else # ellipse/hyperbola
>              parabol := false;
>              conic2(a,b,c,d,e,f,'ok1','ratpoint','x1','y1');
>              ok := ok1;
>              if ok1 then
>                     X := x1; Y := y1
>              else
>                     ratpoint := fail;
>                     X := fail; Y := fail
>              fi
>           fi;
>
>           if ok1 then
>              simplify(subs({x=x1, y=y1},p)); # test (0 as function value)
>           fi
>
> end:
>
```

```
> parabola := proc(a,b,c,d,e,f,ok,x,y)
>              local dp, fp, x1, y1;
> #
> #            IN :
> #            a,b,c,d,e,f : rational functions over Q.
> #
> #            OUT :
> #            ok : boolean.
> #                 (ok = true)  iff
> #                  a*x^2 + b*x*y + c*y^2 + d*x + e*y + f = 0                (GCE)
> #                 defines an irreducible parabola.
> #            x, y : rational functions over Q.
> #                   (ok = true) implies that x and y satisfy (GCE).
> #
> #           LOCAL :
> #           dp, fp : rational functions over Q.
>
>
>           ok := b^2 = 4*a*c and not (a=0 and c=0) and not (a=0 and d=0)
>                         and not (c=0 and e=0);
>           if not ok then RETURN() fi;
>
>           if a <> 0 then
>                     dp := normal(4*a*e - 2*b*d);
>                     fp := normal(4*a*f - d^2);
>                     if dp <> 0 then
>                               y1 := normal(- fp/dp);
>                               y := y1;
>                               x := normal(- (d+b*y1)/(2*a))
>                     else
>                         ok := false;
>                         RETURN()
>                     fi
>          else # c <> 0
>              dp := normal(4*c*d - 2*b*e);
>              fp := normal(4*c*f - e^2);
>              if dp <> 0 then
>                        x1 := normal(- fp/dp);
>                        x := x1;
>                        y := normal(- (e+b*x1)/(2*c))
>              else
>                  ok := false;
>                  RETURN()
>              fi
>          fi;
> #
> #         Some very special case :
> #
>          if f = 0 then x := 0; y := 0; RETURN() fi;
>
>          RETURN()
>
> end:
>
```

```
> quadres := proc(pol1,pol2,modsqt,psqrt)
>            local t, degp1, degp2, p, i, remain, PolSys, VarList, ElSys, n,
>                  cont, Eqns, g, j, ok, Eq, Sol;
> #
> #          IN :
> #          pol1, pol2 : polynomial over Q.
> #
> #          OUT :
> #          modsqt : boolean.
> #                   (modsqt = true) iff  pol1 R pol2.
> #          psqrt : polynomial over Q.
> #                  (modsqt = true) implies that psqrt^2 = pol1 mod pol2.
> #                  psqrt is undefined otherwise.
> #
> #          LOCAL :
> #          t : undeterminate.
> #          degp1, degp2, i, n, j : integer.
> #          p, remain, g : polynomial.
> #          PolSys, VarList, ElSys, Eqns, Sol : list.
> #          Eq : equation.
> #
>          degp1 := degree(pol1); degp2 := degree(pol2);
>          if degp2 = 0 then
>                   if degp1 > 0 then
>                               modsqt := true; psqrt := 0
>                   else
>                      modsqt := evalb(numtheory[L](pol1,abs(pol2)) = 1);
>                      psqrt := numtheory[msqrt](pol1,abs(pol2))
>                   fi;
>                   RETURN();
>          fi;
>          t := indets(pol2)[1];    #  now pol1, pol2 are from Q[t].
>          p := 0;
> #
> #        becomes ansatz for psqrt of degree deg(pol2) - 1
> #        (see next line).
> #
>          for i from 0 to degp2 - 1 do p := p + cat(v,i)*t^i od;
>          remain := rem(p^2 - pol1,pol2,t);
> #
> #        vanishing of this remainder would make p to
> #        a quadratical residue of pol1 modulo pol2.
> #
```

```
>               PolSys := [coeffs(remain,t)];
> #
> #             corresponding polynomial system in the vi´s.
> #
>               VarList := [seq(cat(v,i),i = 0..degp2-1)];
>               ElSys := grobner[gsolve](PolSys,{},VarList);
> #
> #             Groebner Basis for PolSys with 'nice' properties.
> #
>               n := nops(ElSys); ok := true;
> #
> #             Now we try to find a rational solution for the vi´s.
> #
>               i := 1; cont := true;
>               while cont and i <= n do
>                     Eqns := ElSys[i]; g := 0;
>                     j := 0; ok := true;
>                     while ok and j < degp2 do
>                           Eq := Eqns[j+1];
>                           ok := evalb(indets(Eq) = {cat(v,j)});
>                           if ok then
>                              Sol := roots(Eq);
>                              if Sol <> [] then
>                                           g := g + Sol[1][1]*t^j
>                              else
>                                  ok := false
>                              fi
>                           fi;
>                           j := j + 1
>                     od;
>                     cont := not(ok);
>                     i := i + 1
>               od;
>               modsqt := ok;
>               if ok then
>                     psqrt := g
>               else
>                   psqrt := fail
>               fi;
>
>               RETURN();
>
> end:
>
```

```
> sqfrp := proc(p)
>           local factorlist, factornumber, lc, factorof, exponent,
>                 result, i;
> #
> #         IN :
> #         p : polynomial over Q | rational number | integer.
> #
> #         OUT :
> #         the squarefree part of p as function value.
> #
> #         LOCAL :
> #         factorlist : list.
> #         lc : rational.
> #         factornumber, exponent, i : integer.
> #         factorof, result : polynomial
> #
>         if p = 0 then RETURN(1) fi;
>         if type(p,integer) then
>                             readlib(isqrfree);
>                             factorlist := isqrfree(p)
>         elif type(p,rational) then
>              result := sqfrp(numer(p))/sqfrp(denom(p));
>              RETURN(result);
>         else # polynomial case
>             factorlist := sqrfree(p)
>         fi;
> #       Now factorlist = [ lc(p), [[f(1),e(1)],...,[f(m),e(m)]] ],
> #       where p = lc(p)*f(1)^e(1)*...*f(m)^e(m) is a squarefree
> #       factorization of n. The squarefree part of p can then be ex=
> #       pressed as lc(p)*f(1)^(e(1) mod 2)*...*f(m)^(e(m) mod 2).
>
>         factornumber := nops(factorlist[2]);
>         result := 1;
>         lc := factorlist[1];
>         if abs(lc) <> 1 then lc := sqfrp(lc) fi;
>
>         for i to factornumber do
>             factorof := factorlist[2][i][1];
>             exponent := factorlist[2][i][2];
>             result := expand(result * factorof^(exponent mod 2))
>         od;
>
>         result := normal(result * lc);
>
> end:
>
```

```
> iLSolve := proc(a,b,c,solv,x,y,z)
>             local GenSol, SpecSol, u, v, w, var, i;
> #           IN :
> #           a, b, c : integer.
> #
> #           OUT :
> #           solv : boolean.
> #                   (solv = true) iff a*x^2 + b*y^2 + c*z^2 (LE)
> #                    has a nontrivial integral solution.
> #           x, y, z : integer.
> #                      (solv = true) implies that x, y, z are a
> #                       nontrivial integral solution of (LE).
> #
> #           LOCAL :
> #           GenSol, SpecSol : list.
> #           u, v, w, var : undeterminates.
> #           solvable : boolean.
> #           i : interger.
> #
>           GenSol := isolve(a*u^2 + b*v^2 + c*w^2);
>           solvable := evalb(GenSol <> {u=0, v=0, w=0});
>           solv := solvable;
>           SpecSol := eval(subs({_N1=0, _N2=1, _N3=1},GenSol));
>           for i from 1 to 3 do
>               var := lhs(SpecSol[i]);
>               if var = u then
>                  x := abs(rhs(SpecSol[i]))
>               elif var = v then
>                   y := abs(rhs(SpecSol[i]))
>               else
>                   z := abs(rhs(SpecSol[i]))
>               fi
>           od;
>
>           RETURN()
>
> end:
```

```
> LegendreHelp := proc(a,b,solv,x,y,z)
>                 local ok, r, s, T, A, B, m, X, Y, Z;
> #
> #               IN :
> #               a, b : polynomials over Q.
> #                      a, b are squarefree and
> #                      a R b, b R a, (- a*b/gcd(a,b)^2)  R  gcd(a,b).
>
> #
> #               OUT :
> #               solv : boolean.
> #                      (solv = true) iff there exist nonzero
> #                      polynomials x, y, z over R such that
> #                                a*x^2 + b*y^2 = z^2.
> #               x, y, z : polynomials over R.
> #                      (solv = true) implies that
> #                            a*x^2 + b*y^2 = z^2.
> #                      The polynomials are over Q if possible.
> #
> #               LOCAL :
> #               r, s, T, A, B, m,  X, Y, Z : polynomials over Q.
> #               ok : boolean (here dummy variable).
>
>               readlib(psqrt);
>
>               if degree(a) = 0 and degree(b) = 0 then
>                  iLSolve(a,b,-1,'ok','x','y','z');
>                  solv := ok;
>                  if not ok then
>                     if a > 0 then
>                        x := 1; y := 0; z := sqrt(a); solv := true
>                     elif b > 0 then
>                        x := 0; y := 1; z := sqrt(b); solv := true
>                     fi
>                  fi
>               elif degree(a) = 0 and type(degree(b),odd) then
>                    x := 1;
>                    y := 0;
>                    z := sqrt(a);
>                    solv := true
>               elif degree(a) >= degree(b) then
>                    quadres(b,a,'ok','s');  # now s^2 = b mod a.
>                    T := normal((s^2 - b)/a);
>                    A := sqfrp(T);
>                    m := psqrt(normal(T/A));
>                    LegendreHelp(A,b,'solv','X','Y','Z');
>                    x := normal(A*X*m);
>                    y := normal(s*Y+Z);
>                    z := normal(s*Z+b*Y)
>               else
>                    quadres(a,b,'ok','s'); # now s^2 = a mod b.
>                    T := normal((s^2 - a)/b);
>                    B := sqfrp(T);
>                    m := psqrt(normal(T/B));
>                    LegendreHelp(B,a,'solv','Y','X','Z');
>                    y := normal(B*Y*m);
>                    x := normal(s*X+Z);
>                    z := normal(s*Z+a*X)
>               fi;
>
>               RETURN()
>
> end:
```

```
> LegendreSolve := proc(a,b,c,solvable,x,y,z)
>                  local p1, p2, p3, ok, solv, p, lca, lcb, lcc, X, Y, Z,
>                        v1, v2, v3;
> #
> #                IN :
> #                a, b, c : polynomials over Q.
> #                          a, b, c are nonzero, squarefree, pairwise
> #                          relatively prime.
> #
> #                OUT :
> #                solvable : boolean.
> #                x, y, z : polynomials over Q.
> #                          (solvable = true) iff
> #                              a*x^2 + b*y^2 + c*z^2 = 0      (LE)
> #                          has a nontrivial polynomial solution.
> #                          If so, then (x,y,z) is one with z <> 0.
> #                          The polynomials are over Q if possible.
> #
> #                LOCAL :
> #                ok : boolean.
> #                lca, lcb, lcc : integer.
> #                p, p1, p2, p3, X, Y, Z : polynomial.
> #                v1, v2, v3 : indeterminates.
> #
>
> #
> #              Rational Case :
> #
>                if type(a,rational) and type(b,rational)
>                                    and type(c,rational) then
>                                    iLSolve(a,b,c,'solvable','x','y','z');
>                                    RETURN();
>                fi;
> #
> #              Make it easy to reject :
> #              (Sort a,b,c in decreasing order).
> #
```

```
>               if degree(c) <= min(degree(a),degree(b)) then
>                  p3 := c; v2 := 'Z';
>                  if degree(b) <= degree(a) then
>                     p2 := b; p1 := a;
>                     v1 := 'Y'; v3 := 'X'
>                  else
>                      p2 := a; p1 := b;
>                      v1 := 'X'; v3 := 'Y'
>                  fi;
>               elif degree(b) <= min(degree(a),degree(c)) then
>                    p3 := b; v2 := 'Y';
>                    if degree(a) <= degree(c) then
>                       p2 := a; p1 := c;
>                       v1 := 'X'; v3 := 'Z'
>                    else
>                        p2 := c; p1 := a;
>                        v1 := 'Z'; v3 := 'X';
>                    fi
>               else
>                  p3 := a; v2 := 'X';
>                  if degree(b) <= degree(c) then
>                     p2 := b; p1 := c;
>                     v1 := 'Y'; v3 := 'Z';
>                  else
>                      p2 := c; p1 := b;
>                      v1 := 'Z'; v3 := 'Y'
>                  fi
>               fi;
>
>               quadres(normal(-p1*p2),p3,'ok','p');
>               if not ok then solvable := ok; RETURN() fi;
>               quadres(normal(-p1*p3),p2,'ok','p');
>               if not ok then solvable := ok; RETURN() fi;
>               quadres(normal(-p2*p3),p1,'ok','p');
>               solvable := ok;
>               if not ok then RETURN() fi;
>
>               LegendreHelp(normal(-p2*p1),normal(-p3*p1),'ok',v1,v2,v3);
>
>               if p1 = a then X := normal(X/p1)
>               elif p1 = b then Y := normal(Y/p1)
>               else Z := normal(Z/p1)
>               fi;
>
>               x := X; y := Y; z := Z;
>               solvable := ok;
>
>               RETURN()
>
> end:
>
```

```
> conic2 := proc(a,b,c,d,e,f,ok,ratpoint,X,Y)
>            local D, K, L, k1, k2, l1, l2, g, a1, a2, b1, b2, c1, c2, r1, r2,
>                  r3, g1, g2, g3, x, y, y1, z, ratp, ok1;
> #
> #          IN :
> #          a, b, c, d, e, f : rational functions over Q.
> #                             They define a "conic" via
> #          g(x,y) = a*x^2 + b*x*y + c*y^2 + d*x + e*y + f = 0.  (GCE)
> #
> #          OUT :
> #          ok, ratpoint : boolean.
> #          X, Y         : rational functions over R (or fail).
> #                          (ok = true) iff (GCE) defines an
> #                           irreducible ellipse or hyperbola.
> #                          (ok = ratpoint = true) implies that (X,Y) are
> #                           rational functions (over Q if possible) on the
> #                           conic.
> #                          (ok = true; ratpoint = false) implies that
> #                           there is no rational function on the conic and
> #                           that (X,Y) = (fail,fail).
> #
> #          LOCAL :
> #          D, K, L : rational functions over Q.
> #          x, y, z : rational functions over R.
> #          k1, k2, l1, l2, g, a1, a2, b1, b2, c1, c2,
> #          r1, r2, r3, g1, g2, g3 : polynomials over Q.
> #          ratp, ok1 : boolean.
> #
>          readlib(psqrt);
>          D := normal(4*a*c - b^2);
>          ok1 := evalb(D <> 0);
>          if not ok1 then ok := ok1; RETURN() fi;
> #
> #        Transformation of (GCE) in dependence of the values of a and c.
> #
>          if a = 0 and c = 0 then
>                K := -1;
>                L := normal(4*d*e - 4*b*f)
>          elif c <> 0 then
>               K := D;
>               L := normal(4*c^2*d^2 - 4*b*c*d*e + 4*a*c*e^2 +
>                                      4*b^2*c*f - 16*a*c^2*f)
>
>          else # if a<>0 and c=0
>               K := D;
>               L := normal(4*a^2*e^2 - 4*b*a*d*e + 4*b^2*a*f)
>          fi;
> #
> #        Now (GCE) is equivalent to
> #                 x^2 + K*y^2 = L.                        (TE)
> #
```

```
>            ok1 := evalb(L <> 0);  # (degenerate case)
>            ok := ok1;
>            if not ok1 then RETURN() fi;
> #
> #          Some very special case :
> #
>            if f = 0 then ratpoint := true; X := 0; Y := 0; RETURN() fi;
>
>            k1 := numer(K); k2 := denom(K);
>            l1 := numer(L); l2 := denom(L);
>            g := gcd(k2,l2);
>            a1 := normal(l2*k2 / g);
>            b1 := normal(k1*l2 / g);
>            c1 := normal(- l1*k2 / g);
>
>            ok1 := evalb(not( sign(lcoeff(a1)) = sign(lcoeff(b1))
>                         and  sign(lcoeff(b1)) = sign(lcoeff(c1)) ));
>            ok := ok1;
>            if not ok1 then RETURN() fi;
> #
> #          Now (TE) is equivalent to the diophantine equation
> #                   a1*x^2 + b1*y^2 + c1*z^2 = 0.      (DE)
> #
>            a2 := sqfrp(a1); r1 := psqrt(normal(a1/a2));
>            b2 := sqfrp(b1); r2 := psqrt(normal(b1/b2));
>            c2 := sqfrp(c1); r3 := psqrt(normal(c1/c2));
>            g := gcd(a2,gcd(b2,c2));
>            a2 := normal(a2/g); b2 := normal(b2/g); c2 := normal(c2/g);
>            g1 := gcd(a2,b2);
>            a2 := normal(a2/g1); b2 := normal(b2/g1); c2 := normal(c2*g1);
>            g2 := gcd(a2,c2);
>            a2 := normal(a2/g2); b2 := normal(b2*g2); c2 := normal(c2/g2);
>            g3 := gcd(b2,c2);
>            a2 := normal(a2*g3); b2 := normal(b2/g3); c2 := normal(c2/g3);
```

```
> #
> #         Here, (DE) is equivalent to
> #               a2*(x')^2 + b2*(y')^2 + c2*(z')^2 = 0,
> #         where x' = x * r1 / g3, y' = y * r2 / g2, z' = z * r3 / g1.
> #         In addition, a2, b2, c2 satisfy the input-requirements of
> #         LegendreSolve.
> #
>
>         LegendreSolve(a2,b2,c2,'ratp','x','y','z');
>         ratpoint := ratp;
>         if ratp and z = 0 then  # avoid z = 0
>                z := 1;
>                if x <> 0 then
>                   y := normal(y*(1+c2/(4*a2*x^2)));
>                   x := normal(x*(1-c2/(4*a2*x^2)))
>                else
>                   x := normal(x*(1+c2/(4*b2*y^2)));
>                   y := normal(y*(1-c2/(4*b2*y^2)))
>                fi;
>         fi;
>         if ratp then  # we arrive at a rational solution for (TE).
>                x := normal(x * g3 / r1);
>                y := normal(y * g2 / r2);
>                z := normal(z * g1 / r3);
>                x := normal(x / z); y := normal(y / z)
>         else
>             if type(K,rational) and type(L,rational) then
>                if L > 0 then
>                   x := sqrt(L); y := 0
>                else
>                    x := 0; y := sqrt(L/K);
>                fi
>             else
>                X := fail; Y := fail;
>                RETURN()
>             fi
>         fi;
> #
> #       Retransformation
> #
>
>         if a = 0 and c = 0 then
>                X := normal((x + y - 2*e) / (2*b));
>                Y := normal((x - y - 2*d) / (2*b))
>         elif c <> 0 then
>              x := normal((x - 2*d*c + b*e) / K);
>              Y := normal((y - b*x - e) / (2*c));
>              X := x
>         else
>              y1 := normal((x - 2*e*a + b*d) / K);
>              X := normal((y - b*y1 - d) / (2*a));
>              Y := y1
>         fi;
>
>         RETURN()
>
> end:
>
```

```
>
> conic := proc(p,var,ok,parabol,ratpoint,X,Y)
>          local a, b, c, d, e, f, ok1, x1, y1, x, y;
> #
> #        IN :
> #        p : polynomial[var[1],var[2]] with coefficients from Q(t),
>
> #            for some indeterminate t. deg(p) := 2.
> #        var : list of the form [var1, var2], where var1, var2 are the 2
> #              undeterminates occuring in p.
> #
> #        OUT :
> #        ok, parabol, ratpoint : boolean.
> #        X, Y : From R(t) or "fail".
> #               (ok = true) implies that g(x,y) = 0 defines an irreducible
> #                conic with at least two real functions on it.
> #               (ok = parabol = true) iff g(x,y) = 0 defines a parabola.
> #               (ok = ratpoint = true) implies that (X,Y) are the
> #                coordinates of a rational function (over Q if possible)
> #                on the conic.
> #               (ok = true; ratpoint = false) implies that there is no
> #                rational function on the conic.
> #        g(X,Y) as function value (verification : it has to be 0).
> #
> #        LOCAL :
> #        a, b, c, d, e, f : rational functions over Q.
> #        ok1 : boolean.
> #        x1, x2 : rational functions over R or fail.
> #        x, y : undeterminates.
> #
>        x := var[1]; y := var[2];
>        a := normal(coeff(p,x^2));
>        b := normal(coeff(coeff(p,x),y));
>        c := normal(coeff(p,y^2));
>        d := normal(coeff(coeff(p,x),y,0));
>        e := normal(coeff(coeff(p,y),x,0));
>        f := normal(coeff(coeff(p,x,0),y,0));
```

```
> #
> #        Now p = a*x^2 + b*x*y + c*z^2 + d*x + e*y + f.
> #
>          if b^2 = 4*a*c then  # parabolic case
>             parabol := true;
>             parabola(a,b,c,d,e,f,'ok1','x1','y1');
>             ok := ok1;
>             if ok1 then
>                ratpoint := true;
>                X := x1; Y := y1
>             else
>                ratpoint := fail; # interpret fail as "does not matter"
>                X := fail; Y := fail
>             fi
>          else # ellipse/hyperbola
>             parabol := false;
>             conic2(a,b,c,d,e,f,'ok1','ratpoint','x1','y1');
>             ok := ok1;
>             if ok1 then
>                X := x1; Y := y1
>             else
>                ratpoint := fail;
>                X := fail; Y := fail
>             fi
>          fi;
>
>          if ok1 and x1 <> fail then
>             simplify(subs({x=x1, y=y1},p));  # test (0 as function value)
>          fi
>
> end:
>
```

Bibliography

[BOREVICH, SHAFAREVICH 66] "Number Theory" by Z. I. Borevich and I. R. Shafarevich. Academic Press, 1966.

[GEBAUER 91] "Implementation and Analysis of Parametrization Algorithms" by Richard Gebauer (Diploma Thesis). RISC-Linz Report Series No. 91-62, 1991.

[IRELAND, ROSEN 82] "A classical Introduction to modern Number Theory" by K. Ireland and M. Rosen. Springer Verlag, 1982.

[KRAETZEL 81] "Zahlentheory" by E. Krätzel. VEB Dt. Verlag der Wissenschaften, 1981.

[MORDELL 69] "Diophantine Equations" by L. J. Mordell. Academic Press, 1969.

[ROSE 88] "A Course in Number Theory" by H. E. Rose. Oxford Science Publications, 1988.

[SCHARLAU, OPOLKA 84] "From Fermat to Minkowski" by W. Scharlau and H. Opolka. Springer Verlag, 1984.

[SENDRA, WINKLER 91] "Symbolic Parametrization of Curves", by J. R. Sendra and F. Winkler. Journal of Symbolic Computation 12/6, p. 607 - 631 (1991).

[SENDRA, WINKLER 96] "Parametrization of Algebraic Curves over Optimal Field Extensions", by J. R. Sendra and F. Winkler. Journal of Symbolic Computation, to appear (1996).