The Algorithmic Solution of Simultaneous Diophantine Equations

Rachel Louise Long

Department of Mathematical Sciences

Oxford Brookes University

Thesis submitted in partial fulfilment of the requirements

of the award of Doctor of Philosophy

as awarded by Oxford Brookes University

July 2005

Abstract

A new method is presented for solving pairs of simultaneous Diophantine equations, such as those which result from the 2-descent process on elliptic curves. The method works by determining a set of solutions modulo a prime P, raising each of these solutions to a set of P solutions modulo P^2, and then determining a solution modulo P^6 for each of the solutions modulo P^2. These solutions modulo P^6 lie on a lattice which is then reduced using a suitable lattice reduction algorithm. The required solution can then be written as a linear combination of the basis vectors for the lattice, and the coefficients in this combination are determined. The running time of this algorithm is $O(N^{2/3})$ where N is a bound on the size of the solution required. Variations on the method are also presented.

Following a 2-descent on elliptic curves of the form $Y^2 = X^3 + pX$, where $p \equiv 5 \pmod 8$ originally described by Bremner and Cassels [8], the methods are applied to various pairs of equations. Generators for the free abelian part of the group of rational points on each of these curves are presented, including the case $p = 16421$ which has a canonical height of 137.2290.

By combining the method with existing techniques, we also find a generator for the set of points of infinite order on the curve $Y^2 = X^3 + 17477X$. This point has canonical height $\hat{h}(P) = 406.4797$.

We also find a generator on the Mordell curve $y^2 = x^3 + 7823$, which is the only case missing from the tables of Gebel, Pethö and Zimmer for the curves $y^2 = x^3 + k$ with $|k| \leq 10000$ [20].

Acknowledgements

I would like to thank my Director of Studies, Dr. Catherine Hobbs, and my supervisors, Prof. Roger Heath-Brown and Prof. Mike Pidcock, for their advice and support. I also acknowledge the financial support of the Engineering and Physical Sciences Research Council.

For Richard, Lottie and Tom.

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Summary of Contents 4

2 Solving Diophantine Equations **6**

2.1 Overview 6

2.2 Congruence Considerations 7

2.3 Local-Global Principle 9

2.4 Quadratic Residues and Quadratic Reciprocity 10

2.5 Factorisation over the Integers 12

2.6 Factorisation over Algebraic Number Fields 14

2.7 Fermat's Method of Descent 16

3 Elliptic Curves over $\mathbb{Q}$ **19**

3.1 Overview 19

3.2 The group $E(\mathbb{Q})$ 30

3.2.1 Computing the Torsion Subgroup 38

3.2.2 Calculating the Rank 39

3.2.3 Finding Generators 58

3.3 The Birch and Swinnerton-Dyer Conjecture 62

3.4 Heegner points 64

4 The Equation $Y^2 = X^3 + pX$, where p is prime **68**

4.1 Overview 68

4.2 Algebraic Preliminaries 70

4.3 The Equation $Y^2 = X^3 + pX$, where $p \equiv 5 \pmod 8$ 76

4.3.1 Finding u and v such that $p = u^2 + 4v^2$ 83

4.3.2 Finding an equivalent quadratic form 86

4.4 Maple Worksheet 91

5 Lattices and Lattice Reduction **92**

5.1 Overview 92

5.2 LLL-reduced Lattice Bases 95

5.3 The LLL Algorithm 98

6 The Solution of Pairs of Simultaneous Quadratic Equations **102**

6.1 Overview 102

6.2 The Elkies method 106

6.3 The `p6sim` method 111

6.4 The `p4quart` method 125

6.5 The `composite` method 127

6.6 Maple Worksheets 130

7 Results and Conclusions **131**

7.1 Overview 131

7.2 Analysis of results 144

7.2.1 The `p6sim` method 144

7.2.2 The `p4quart` method 147

7.2.3 The `p4sim` method 149

7.2.4 The `composite` method 153

7.2.5 Using a smaller auxiliary prime 157

7.2.6 Other quadric intersections 159

7.3 Results for other implementations 164

7.3.1 Naïve searches 164

7.3.2 Heegner point constructions 167

7.3.3 2-descent 168

7.4 Combinations of new and existing methods 175

7.5 Limitations and further work 178

A descent.mws **185**

B p6sim.mws **202**

C p4quart.mws **212**

D p4sim.mws 219

E composite.mws 226

List of Tables

4.1 Possible parities of r, s and t in equation (4.5) 77

4.2 The four possibilities for equation (4.14) 78

7.1 Expected heights of generators 132

7.2 Values of r, s and t for the generators of various elliptic curves of the form $y^2 = x^3 + px$ 142

7.3 Run times (in seconds) for `p6sim`, `p4quart` and `p4sim` 143

7.4 Run times (in seconds) for the `composite` method 154

7.5 Run times (in seconds) using smaller auxiliary primes 158

7.6 Run times (in seconds) for the naïve searches 165

7.7 Points found by a Heegner point construction 168

7.8 Results from `mwrank` 171

7.9 Run times (in seconds) for `Apecs` 173

List of Figures

3.1 The elliptic curve $y^2 = x^3 + 3x + 3$ has $\Delta < 0, A > 0$ 24

3.2 The elliptic curve $y^2 = x^3 - 3x + 3$ has $\Delta < 0, A \leq 0$ 25

3.3 The elliptic curve $y^2 = x^3 - 3x + 1$ has $\Delta > 0$ 26

3.4 A straight line intersects an elliptic curve at three points 31

3.5 Addition of points on an elliptic curve 34

3.6 The multiplication-by-2 map 40

3.7 2-descent via 2-isogeny 43

3.8 General 2-descent 46

3.9 The second descent in the 2-isogeny case 52

3.10 The second descent in the general case 58

7.1 Run times for the `p6sim` method 146

7.2 Run times for the `p4quart` method 148

7.3 Run times for the `p4sim` method 151

7.4 Run times for `p6sim`, `p4quart` and `p4sim` 152

7.5 Run times for `composite` 156

7.6 Run times for the naïve searches 166

7.7 Run times for `apecs` 174

Chapter 1

Introduction

1.1 Overview

Diophantine equations are polynomial equations with integer coefficients in two or more variables, for which we seek solutions in integers, natural numbers or rational numbers. Since a set of m equations f_i in $n \geq 2$ variables describes an affine variety V in n-dimensional affine space $\mathbb{A}^n$, determining the integer or rational solutions $(x_1, x_2, ..., x_n)$ to these equations is equivalent to finding the integer or rational points $V(\mathbb{Z})$ or $V(\mathbb{Q})$ on the corresponding variety.

The general equations representing affine varieties are inhomogeneous - the sums of the powers of the variables may differ from term to term. By making the substitutions

$$x_1 = \frac{X_1}{X_{n+1}}, \quad x_2 = \frac{X_2}{X_{n+1}}, \quad \ldots, \quad x_n = \frac{X_n}{X_{n+1}}$$

and clearing denominators, we obtain a set of m homogeneous equations in $n + 1$ variables, $F_i(X_1, \ldots, X_{n+1}) = 0$. These equations define a projective variety in $\mathbb{P}^n$. Since the equations are homogeneous, we may search for coprime integer solutions $(X_1 : X_2 : \ldots : X_{n+1})$.

We will concentrate on curves defined over $\mathbb{Q}$. A curve C can be expressed affinely as a Diophantine equation in two variables, $f(x, y) = 0$, or projectively as $F(X, Y, Z) = 0$. Given a curve C over $\mathbb{Q}$, we wish to know whether any rational points lie on C and, if so, whether we can in some way describe these points. The existence, and finiteness, of a set of rational points on C depends on the genus of the curve. The genus of C is defined to be the number of holes in the compact Riemann surface $C(\mathbb{C})$.

Affine curves with genus greater than 1 have degree at least 4, and are not always smooth. Faltings' theorem [19] states that for a curve C of genus $g \geq 2$ defined over $\mathbb{Q}$, $C(\mathbb{Q})$ is finite (and possibly empty). However, Faltings' proof and subsequent proofs by others [5, 50, 29] do not provide an effective algorithm for finding solutions. The proofs allow a bound on the number of rational points on C to be calculated, but do not provide an upper bound on the numerators and denominators of the coordinates.

Using the Riemann-Roch theorem, it can be shown that a curve C of genus 0 is isomorphic to a conic in $\mathbb{P}^2$. Then, by a linear substitution, we find that C can be represented by the equation $F(X, Y, Z) = aX^2 + bY^2 + cZ^2 = 0$ for some non-zero integers a, b, c. It is known that $C(\mathbb{Q})$ is non-empty if and only if $C(\mathbb{R})$ and $C(\mathbb{Q}_p)$ for all primes p are non-empty. This is known as the Hasse principle. Criteria

for deciding whether $\mathbb{Q}_p$-points exist are given in [35]. If $C(\mathbb{Q})$ is non-empty then $C \simeq \mathbb{P}^1$ over $\mathbb{Q}$, so that C can be parameterised by rational functions. In this case, there will be infinitely many rational points [24].

A genus 1 curve C over $\mathbb{Q}$ need not have a rational point, and Lind [26] and Reichardt [38] have given examples of curves C which fail the Hasse principle. If C does have a rational point, then C is an elliptic curve, isomorphic to the projective closure of the affine curve $y^2 = x^3 + Ax + B$ for some integers A and B with $4A^3 + 27B^2 \neq 0$. The set of rational points $C(\mathbb{Q})$ is a finitely generated abelian group. If the curve has rank 0, then $C(\mathbb{Q})$ is finite. Otherwise, there will be infinitely many points in $C(\mathbb{Q})$. While it is relatively easy to find the (finite) set of rational points of finite order on C, determining the rank and a set of generators for the points of infinite order is more difficult. The most commonly used approach is to use the method of descent to produce a covering curve, and then to find points on the covering curve. If higher descents are used we obtain a pair of simultaneous homogeneous quaternary quadratic equations, representing the intersection of two quadrics, and we attempt to find integer solutions to these equations. Given a bound N on the absolute value of the integers, the most naïve search for these solutions would take time $O(N^4)$. A new method of searching for solutions to these equations, with running time $O(N^{2/3})$ is presented in this thesis.

1.2 Summary of Contents

Chapter 2 contains a summary of some of the existing techniques for solving Diophantine equations. Such techniques can be found in many textbooks, including [34], [39] and [23].

Chapter 3 introduces elliptic curves defined over $\mathbb{Q}$, and describes the group of rational points on such curves. This group of points consists of a torsion subgroup of points of finite order, and a free abelian part. Methods for determining the torsion subgroup, finding generators of the free abelian part, and calculating a bound on the rank of the curve are given.

In order to calculate the rank of an elliptic curve, a process known as 2-descent is used. In 1984, Bremner and Cassels [8] gave details of an algorithm for performing the descent on curves of the form $Y^2 = X^3 + pX$, where p is a prime congruent to 5 modulo 8. This descent is described in detail in chapter 4.

Chapter 5 provides a brief summary of lattices and lattice reduction, and includes the LLL algorithm devised by Lenstra, Lenstra and Lovász in 1982 [25].

In chapter 6, a new method is presented for solving pairs of simultaneous equations, such as those which result from the 2-descent process on elliptic curves. This is an improvement on the so-called 'p-adic Elkies method'. The algorithm works by determining a set of solutions modulo a prime P, raising each of these solutions to a set of P solutions modulo P^2, and then determining a solution modulo P^6 for each of the solutions modulo P^2. These solutions modulo P^6 lie on

a lattice which is then reduced using the techniques of chapter 5. The required solution can then be written as a linear combination of the basis vectors for the lattice, and the coefficients in this combination are determined. For the pairs of quaternary equations resulting from the descent process, the running time of this algorithm is $O(N^{2/3})$ where N is a bound on the size of the solution required.[1] Two variations on the algorithm are also given. For comparison, a naive search testing every possible vector (x_1, x_2, x_3, x_4) with $x_i \leq N$ would take time $O(N^4)$, while eliminating one variable from the pair of simultaneous equations and then performing a naive search on the remaining three would take time $O(N^3)$.

The results of running these new algorithms for a variety of elliptic curves are given in chapter 7.

[1]Note that in all run time estimates, lower order terms have been ignored.

Chapter 2

Solving Diophantine Equations

2.1 Overview

Various techniques from elementary and algebraic number theory can be used in order to determine whether an equation has any solutions, and to solve the equation if it does.

A summary of existing techniques follows.

2.2 Congruence Considerations

Congruence considerations are often very revealing, especially when considering equations modulo powers of a prime. The congruence $f(x_1, x_2, ..., x_n) \equiv 0 \pmod{M}$ is equivalent to the Diophantine equation $f(x_1, x_2, ..., x_n) = Mx_{n+1}$. So for the equation $f(x_1, x_2, ..., x_n) = 0$ to have solutions, we require $Mx_{n+1} = 0$ for any M, or in other words we require $x_{n+1} = 0$. If we can find some M for which the congruence $f(x_1, x_2, ..., x_n) \equiv 0 \pmod{M}$ has no solutions, then the Diophantine equation $f(x_1, x_2, ..., x_n) = 0$ can have no integer solutions.

Example: Suppose that $a, b, c \in \mathbb{Z}$, and p is a prime such that $p \nmid abc$. Then the equation

$$ax^3 + bpy^3 + cp^2z^3 = 0 \tag{2.1}$$

has no solutions in integers or rational numbers.

Proof. Suppose that a solution to (2.1) is given by (x_0, y_0, z_0). Since the equation (2.1) is homogeneous, we can assume without loss of generality that $\gcd(x_0, y_0, z_0) = 1$. Since $p \mid bpy_0^3$ and $p \mid cp^2z_0^3$, we must have $p \mid ax_0^3$. But $p \nmid a$ so $p \mid x_0^3$. Since p is prime (so that in particular p is not a cube), we must have $p \mid x_0$. Now we divide through by p to give

$$\frac{ax_0^3}{p} + by_0^3 + cpz_0^3 = 0,$$

which can be rewritten as

$$ap^2\left(\frac{x_0}{p}\right)^3 + by_0^3 + cpz_0^3 = 0,$$

or

$$by_0^3 + cpz_0^3 + ap^2\left(\frac{x_0}{p}\right)^3 = 0.$$

The argument can be repeated, so that $p \mid y_0$. Then

$$bp^2\left(\frac{y_0}{p}\right)^3 + cz_0^3 + ap\left(\frac{x_0}{p}\right)^3 = 0,$$

or

$$cz_0^3 + ap\left(\frac{x_0}{p}\right)^3 + bp^2\left(\frac{y_0}{p}\right)^3 = 0.$$

Repeating the argument a further time gives $p \mid z_0$, contradicting the assumption that $\gcd(x_0, y_0, z_0) = 1$. □

2.3 Local-Global Principle

We have seen that if there is some modulus M for which the congruence $f(x_1, x_2, ..., x_n) \equiv 0 \pmod{M}$ has no solutions, then integer solutions cannot exist. In this case, the integer solutions are called 'global' solutions, while the solutions modulo M are called 'local' solutions. For the existence of global solutions in $\mathbb{Q}$ we require local solutions in all local completions of $\mathbb{Q}$, namely solutions modulo M for all M and solutions in $\mathbb{R}$.

An equation for which the existence of all local solutions *guarantees* the existence of global solutions is said to satisfy the Hasse principle (also known as the local-global principle). However, the Hasse principle does not hold for all equations. For example, Selmer [41] has shown that the equation $3X^3 + 4Y^3 + 5Z^3 = 0$ has solutions modulo M for all M, but no non-trivial integer solutions.

2.4 Quadratic Residues and Quadratic Reciprocity

If p is an odd prime and $p \nmid a$, then we say that a is a *quadratic residue* modulo p if the equation $x^2 = a \pmod{p}$ is solvable; if the equation is not solvable, then we say that a is a *quadratic non-residue* modulo p. This is denoted by the Legendre symbol

$$\left(\frac{a}{p}\right) = \begin{cases} +1 & \text{if } a \text{ is a quadratic residue mod } p \\ -1 & \text{if } a \text{ is a quadratic non-residue mod } p \\ 0 & \text{if } p \mid a. \end{cases}$$

Useful properties of the Legendre symbol are given by the following three theorems.

Theorem 2.4.1. *For p an odd prime, we have the following results.*

(a) *If $a \equiv b \pmod{p}$, then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$.*

(b) *For any integers a and b, $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$.*

(c) $$\left(\frac{-1}{p}\right) = \begin{cases} +1 & \text{if } p \equiv 1 \pmod{4} \\ -1 & \text{if } p \equiv 3 \pmod{4}. \end{cases}$$

(d) $$\left(\frac{2}{p}\right) = \begin{cases} +1 & \text{if } p \equiv 1, 7 \pmod{8} \\ -1 & \text{if } p \equiv 3, 5 \pmod{8}. \end{cases}$$

Theorem 2.4.2 (Euler's Criterion). *If $p \nmid a$, then*

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

Theorem 2.4.3 (Gauss's Law of Quadratic Reciprocity). *If p and q are distinct*

odd primes, then

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}},$$

or equivalently

$$\left(\frac{p}{q}\right) = \begin{cases} -\left(\frac{q}{p}\right) & p \equiv q \equiv 3 \pmod 4 \\ \left(\frac{q}{p}\right) & \textit{otherwise.} \end{cases}$$

These properties can be used in the solution of quadratic Diophantine equations.

Example: The equation

$$y^2 = 41x + 3 \tag{2.2}$$

has no solution in integers.

Proof. Suppose the equation has an integer solution. Reduction modulo 41 gives that $y^2 \equiv 3 \pmod{41}$, so $\left(\frac{3}{41}\right) = 1$. But the properties of the Legendre symbol and the law of quadratic reciprocity gives $\left(\frac{3}{41}\right) = \left(\frac{41}{3}\right) = \left(\frac{2}{3}\right) = -1$ by theorem 2.4.1(d). Therefore there are no solutions to the equation (2.2) in integers. □

2.5 Factorisation over the Integers

It may be possible to factorise one side of an equation over the integers. Then we can use other techniques to show that no solution can exist. We demonstrate this method by looking at a specific example of the Mordell equation

$$y^3 = x^2 + k,$$

where $k \in \mathbb{Z}$.

Example: The equation

$$y^3 = x^2 - 7 \tag{2.3}$$

has no solutions in $\mathbb{Z}$.

Proof. By taking residues modulo 4, we see that if y is even then $y^3 \equiv 0 \pmod 4$, so that $x^2 - 7 \equiv x^2 - 3 \equiv 0 \pmod 4$ so $x^2 \equiv 3 \pmod 4$. But the squares modulo 4 are 0 (for an even number) and 1 (for an odd number), so there are no solutions if y is even. Therefore, y must be odd.

We can add 8 to each side of the equation (2.3) to give $x^2 + 1 = y^3 + 8$, which we can factorise as $x^2 + 1 = (y + 2)(y^2 - 2y + 4)$. Since y is odd, we have $y^2 \equiv 1$ and $-2y \equiv 2 \pmod 4$, so that $(y^2 - 2y + 4) \equiv 3 \pmod 4$. Any number congruent to 3 (mod 4) has a prime factor $p \equiv 3 \pmod 4$. So we must have $p \mid (y^2 - 2y + 4) \Rightarrow p \mid (y + 2)(y^2 - 2y + 4) \Rightarrow p \mid x^2 + 1$. So $x^2 \equiv -1 \pmod p$. But this is impossible for any prime p such that $p \equiv 3 \pmod 4$ by theorem 2.4.1(c). Therefore, there

are no integer solutions. □

2.6 Factorisation over Algebraic Number Fields

We have seen in the previous subsection that the Mordell equation has no integer solutions when $k = -7$, and many other values of k can be treated similarly. However, if $y^3 - k + 1$ cannot be factorised over the integers, we must consider other number fields. The following example illustrates this technique for $k = 2$.

Example: The equation

$$y^3 = x^2 + 2 \tag{2.4}$$

has only the integer solutions $x = \pm 5, y = 3$.

Proof. If x is even, then $x^2 + 2$ is even, so that y^3 is even. This would imply that y is even, so that the left-hand side of (2.4) is divisible by 8. This would be impossible since the right-hand side of (2.4) is only divisible by 2. Hence x must be odd.

We work in the unique factorisation domain $\mathbb{Z}\left[\sqrt{-2}\right]$. The integers $\mathbb{Z}\left[\sqrt{-2}\right]$ consist of numbers of the form $a + b\sqrt{-2}$ with $a, b \in \mathbb{Z}$. This gives

$$y^3 = (x + \sqrt{-2})(x - \sqrt{-2}). \tag{2.5}$$

Suppose that these two factors had a common factor $c + d\sqrt{-2}$. Then this common factor must also divide their sum and difference, so that $c + d\sqrt{-2} \mid 2x$ and $c + d\sqrt{-2} \mid 2\sqrt{-2}$. Taking norms, we have

$$c^2 + 2d^2 \mid 4x^2, \quad c^2 + 2d^2 \mid 8,$$

and hence $c^2 + 2d^2 \mid 4$. The only possibilities are $c = \pm 1, d = 0$, or $c = \pm 2, d = 0$. Since none of these are proper factors of $x + \sqrt{-2}$, the two factors in the right-hand side of (2.5) must be coprime.

Since the units in $\mathbb{Z}\left[\sqrt{-2}\right]$ are ± 1, both of which are cubes, we have

$$\begin{aligned} x + \sqrt{-2} &= \left(a + b\sqrt{-2}\right)^3 \\ &= \left(a^3 - 6ab^2\right) + \left(3a^2b - 2b^3\right)\sqrt{-2}, \end{aligned} \tag{2.6}$$

for some $a, b \in \mathbb{Z}$.

Comparing coefficients of $\sqrt{-2}$ in (2.6), we have

$$1 = \left(3a^2b - 2b^3\right),$$

for which the only integer solutions are $a = \pm 1, b = 1$.

Then

$$\begin{aligned} x &= a^3 - 6ab^2 \\ &= \begin{cases} -5 & \text{if } a = 1 \\ 5 & \text{if } a = -1. \end{cases} \end{aligned}$$

This gives $y^3 = 27$, so that $y = 3$. □

2.7 Fermat's Method of Descent

Assume that a solution to our equation can be found, and that in some way we can take a "smallest" solution. Fermat's method of descent then uses this solution to derive a smaller one, contradicting the assumption that the original solution was the smallest possible.

Example: The equation

$$x^4 + y^4 = z^2 \tag{2.7}$$

has no integer solutions with $xyz \neq 0$.

Proof. Suppose that such a solution (x, y, z) exists. Since the powers in equation (2.7 are all even, we may assume that $x, y, z > 0$. If $\gcd(x, y, z) = d$, then we have $d^4 \mid x^4 + y^4 = z^2$, so that $d^2 \mid z$. We can then replace x, y, z by $\frac{x}{d}, \frac{y}{d}, \frac{z}{d^2}$ respectively to obtain a solution in coprime integers. So we may assume $\gcd(x, y, z) = 1$, and that z is the smallest possible.

Congruence modulo 4 shows that z must be odd, and that x and y are of opposite parity. So without loss of generality, we may assume that x is odd and y is even.

Rearranging (2.7), we have

$$y^4 = z^2 - x^4 = (z - x^2)(z + x^2).$$

Suppose p is a prime with $p \mid (z - x^2)$ and $p \mid (z + x^2)$. Then $p \mid 2z$ and $p \mid 2x^2$, and

since $\gcd(x, z) = 1$ we have $p \mid 2$. Since x and z are both odd, $(z - x^2)$ and $(z + x^2)$ are both even. Therefore $\gcd(z - x^2, z + x^2) = 2$.

Now, y is even, so $2^4 \mid y^4 = (z - x^2)(z + x^2)$. But $\gcd(z - x^2, z + x^2) = 2$, so 2 must divide one factor exactly and 2^3 must divide the other.

There are two cases to consider. Either

$$\begin{cases} z - x^2 = 2a^4 & \text{where } a > 0, a \text{ odd} \\ z + x^2 = 8b^4 & \text{where } b > 0, \gcd(a, b) = 1, \end{cases} \tag{2.8}$$

or

$$\begin{cases} z - x^2 = 8b^4 & \text{where } b > 0 \\ z + x^2 = 2a^4 & \text{where } a > 0, a \text{ odd}, \gcd(a, b) = 1. \end{cases} \tag{2.9}$$

In the first case, we can eliminate z from (2.8) to obtain $x^2 = 4b^4 - a^4 \equiv 0 - 1 = 3 \pmod 4$. But x is odd, so $x^2 \equiv 1 \pmod 4$. Hence there are no solutions in this case, and the second case must hold.

Eliminating x from (2.9) in the second case, we obtain $z = a^4 + 4b^4$. Rearranging this, we have $a^4 = z - 4b^4$. Since $4b^4 > 0$, we have $a^4 < z \Rightarrow a < z$.

Eliminating z from (2.9) gives $x^2 = a^4 - 4b^4$, so that $4b^4 = a^4 - x^2 = (a^2 - x)(a^2 + x)$. Again, $\gcd(a^2 - x, a^2 + x) = 2$. Hence

$$\begin{aligned} a^2 - x &= 2c^4 \\ a^2 + x &= 2d^4, \end{aligned} \tag{2.10}$$

for some $c, d \in \mathbb{Z}$. Eliminating x from (2.10), we obtain

$$a^2 = c^4 + d^4, \tag{2.11}$$

so that (c, d, a) is a solution to (2.7). But $a < z$, contradicting the assumption that (x, y, z) is the solution with smallest possible z. Hence there can be no non-trivial integer solutions. □

Chapter 3

Elliptic Curves over $\mathbb{Q}$

3.1 Overview

Definition 3.1.1. An *elliptic curve* defined over $\mathbb{Q}$ is a projective plane curve defined affinely by the equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{3.1}$$

with $a_i \in \mathbb{Q}$, together with a single point at infinity, O_E.

The form of equation (3.1) is known as *long Weierstrass form*, but a more simple equation can be used.

Theorem 3.1.2. *The curve (3.1) is equivalent to a curve of the form*

$$y^2 = x^3 + Ax + B, \tag{3.2}$$

with $A, B \in \mathbb{Z}$.

Proof. Multiplying each side of (3.1) by 4 gives

$$4y^2 + 4a_1xy + 4a_3y = 4x^3 + 4a_2x^2 + 4a_4x + 4a_6.$$

We then complete the square on the left hand side

$$(2y + a_1x + a_3)^2 - a_1^2x^2 - 2a_1a_3x - a_3^2 = 4x^3 + 4a_2x^2 + 4a_4x + 4a_6,$$

so that

$$(2y + a_1x + a_3)^2 = 4x^3 + (a_1^2 + 4a_2)x^2 + (2a_1a_3 + 4a_4)x + (a_3^2 + 4a_6).$$

Let $Y' = 2y + a_1x + a_3$. Then

$$(Y')^2 = 4x^3 + b_2x^2 + 2b_4x + b_6,$$

where

$$b_2 = a_1^2 + 4a_2$$

$$b_4 = a_1a_3 + 2a_4 \tag{3.3}$$

$$b_6 = a_3^2 + 4a_6.$$

Now let

$$x = \frac{x' - 3b_2}{36}$$

$$Y' = \frac{y'}{108}.$$

Then (on multiplying through by 11664) we obtain

$$\begin{aligned}(y')^2 &= (x')^3 + (648b_4 - 27b_2^2)x' + (54b_2^3 - 1944b_2b_4 + 11664b_6)\\ &= (x')^3 - 27(b_2^2 - 24b_4)x' - 54(-b_2^3 + 36b_2b_4 - 216b_6)\\ &= (x')^3 - 27c_4x' - 54c_6,\end{aligned}$$

where

$$c_4 = b_2^2 - 24b_4$$

$$c_6 = -b_2^3 + 36b_2b_4 - 216b_6. \tag{3.4}$$

The coefficients $-27c_4$ and $-54b_6$ are rational. However, we can restrict the coefficients to be integers, as follows.

Suppose that

$$-27c_4 = \frac{m}{n},$$

$$-54c_6 = \frac{p}{q}.$$

This gives

$$(y')^2 = (x')^3 + \frac{m}{n}x' + \frac{p}{q},$$

so that

$$n^6q^6(y')^2 = n^6q^6(x')^3 + n^5q^6mx' + pn^6q^5,$$

or

$$(n^3q^3y')^2 = (n^2q^2x')^3 + n^3q^4m(n^2q^2x') + pn^6q^5.$$

Finally, let

$$X = n^2q^2x',$$

$$Y = n^3q^3y'.$$

Then

$$Y^2 = X^3 + AX + B$$

where

$$A = n^3q^4m,$$

$$B = pn^6q^5$$

are integers. □

This form of equation is known as *short Weierstrass form*.

Definition 3.1.3. For a curve of the form (3.1) with the quantities b_2, b_4, b_6, c_4 and c_6 defined by (3.3) and (3.4), the *discriminant*, Δ, is given by

$$\Delta = \frac{c_4^3 - c_6^2}{1728}.$$

The discriminant of the curve (3.2) is given by

$$\Delta = -16(4A^3 + 27B^2).$$

We can also define the quantity

$$b_8 = a_1^2 a_6 + 4a_2 a_6 - a_1 a_3 a_4 + a_2 a_3^2 - a_4^2. \tag{3.5}$$

The curve is non-singular if and only if $\Delta \neq 0$. Figures 3.1, 3.2 and 3.3 demonstrate how the sign of the discriminant affects the shape of the graph.

A related property of an elliptic curve is the j-invariant, $j = \frac{c_4^3}{\Delta}$. The j-invariant classifies elliptic curves up to isomorphism over the complex numbers (see, for example, [43]).

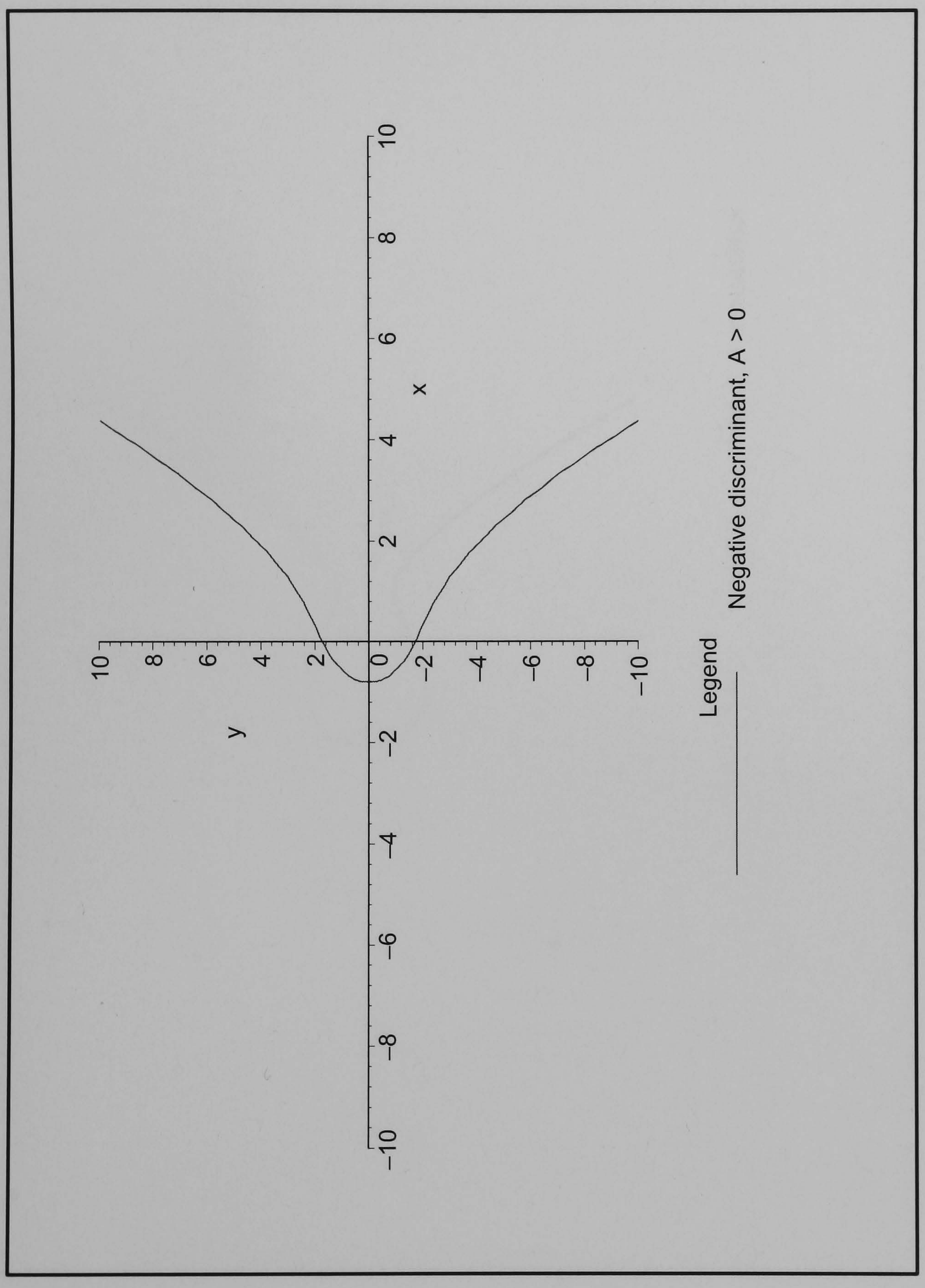

Figure 3.1: The elliptic curve $y^2 = x^3 + 3x + 3$ has $\Delta < 0, A > 0$

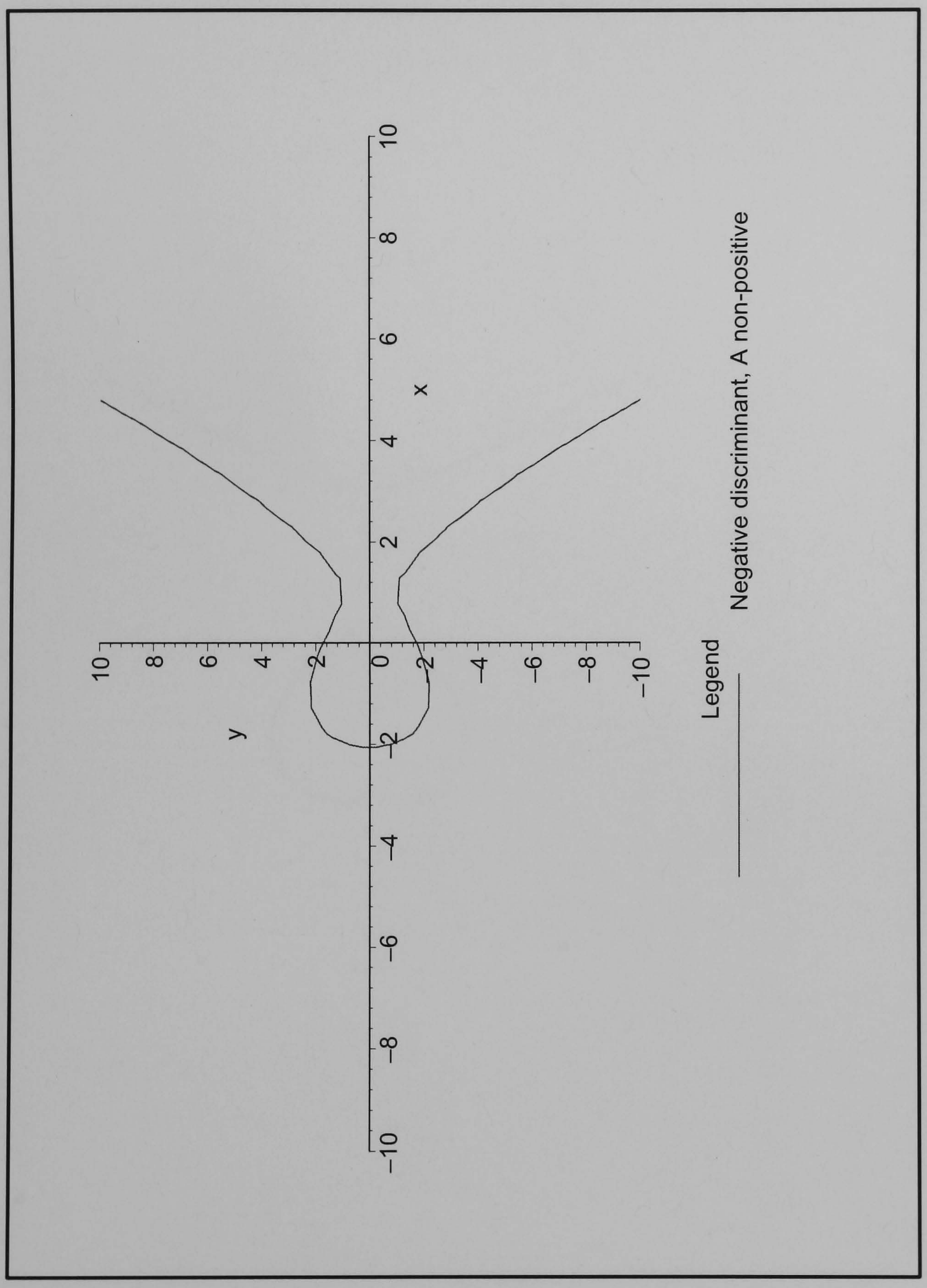

Figure 3.2: The elliptic curve $y^2 = x^3 - 3x + 3$ has $\Delta < 0, A \leq 0$

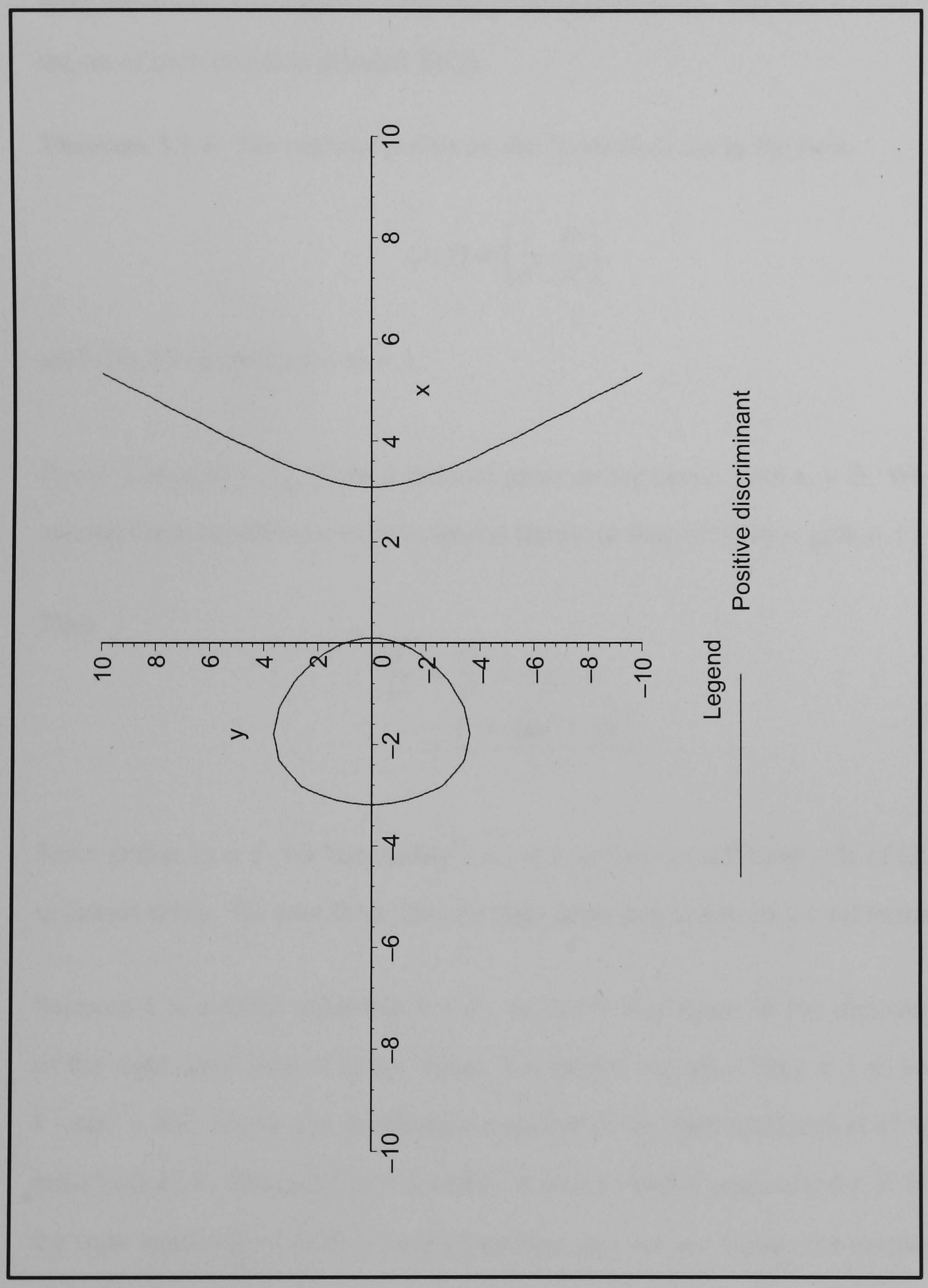

Figure 3.3: The elliptic curve $y^2 = x^3 - 3x + 1$ has $\Delta > 0$

We are interested in the rational points on this curve, i.e. pairs $(x, y) \in \mathbb{Q}^2$ satisfying equation (3.2), together with the point at infinity O_E. For any such curve E, the set of such points is denoted $E(\mathbb{Q})$.

Theorem 3.1.4. *The rational points on the curve (3.2) are of the form*

$$(x, y) = \left(\frac{l}{n^2}, \frac{m}{n^3}\right),$$

with $l, m, n \in \mathbb{Z}$ *and* $\gcd(l, n) = 1$.

Proof. Let $(x, y) = \left(\frac{l}{u}, \frac{m}{v}\right)$ be a rational point on the curve, with $u > 0$. We may assume these coordinates to be in lowest terms, so that $\gcd(l, u) = \gcd(m, v) = 1$. Then

$$\begin{aligned} \frac{m^2}{v^2} &= \frac{l^3}{u^3} + A\frac{l}{u} + B \\ &= \frac{l^3 + Alu^2 + Bu^3}{u^3}. \end{aligned} \tag{3.6}$$

Since $\gcd(m, v) = 1$, we have $\gcd(m^2, v^2) = 1$ so that the left hand side of (3.6) is in lowest terms. We now show that the right hand side is also in lowest terms.

Suppose k is a prime such that $k \mid u^3$, so that k is a factor of the denominator of the right hand side of (3.6). Since k is prime, we must have $k \mid u$, so that $k \mid Alu^2 + Bu^3$. So for k to divide the numerator of the right hand side of (3.6), we must have $k \mid l^3$. But $\gcd(l, u) = 1$ and $k \mid u$, so $k \nmid l$ and in particular $k \nmid l^3$. Hence the right hand side of (3.6) is in lowest terms, and we can equate the numerators and denominators.

Equating denominators, we have $v^2 = u^3$, so that v^2 is both a square and a cube. Similarly, u^3 is both a cube and a square. Hence each must be a sixth power.

Then

$$v^2 = n^6 \Rightarrow v = n^3,$$

$$u^3 = n^6 \Rightarrow u = n^2.$$

Finally, since $\gcd(l, n^2) = \gcd(l, u) = 1$, we have $\gcd(l, n) = 1$ as required. □

In order to measure the "size" of a rational point, we use the height function. Several versions exist.

Definition 3.1.5. The *(naïve) height of a rational number* $x = \frac{m}{n}$ with $\gcd(m, n) = 1$ is given by

$$H(x) = \max(|m|, |n|).$$

We define the *(naïve) height of a rational point* $P = (x, y)$, with $x = \frac{l}{n^2}$ in lowest terms to be

$$H(P) = H(x) = \max(|l|, n^2).$$

The *(logarithmic) height* (or Weil height), $h(P)$, is defined to be

$$h(P) = \log H(P).$$

We define $H(O_E) = 1$ and $h(O_E) = 0$.

A modified version of the height of a point on an elliptic curve is given by the *canonical height* (or Neron-Tate height), $\hat{h}(P) = \lim_{n\to\infty} 4^{-n} h(2^n P)$, which can be

taken to be an approximation to the logarithmic height. Faster methods for computing the canonical height of a point on an elliptic curve exist (see, for example, [12]).

The difference between the logarithmic and canonical height of a rational point on an elliptic curve is bounded [55, 44].

Lemma 3.1.6. *There exist constants* c_1, c_2*, depending on the curve* E *but not on the point* P*, such that*

$$-c_1 \leq \hat{h}(P) - h(P) \leq c_2.$$

Silverman [44] gives the values of the constants c_1 and c_2 as

$$c_1 = \frac{1}{12}h(j) + \mu(E) + 1.946,$$

$$c_2 = \mu(E) + 2.14,$$

where

$$\mu(E) = \frac{\log|\Delta| + \log^+(j)}{6} + \log^+\left(\frac{b_2}{12}\right) + \log(2^*),$$

$$\log^+(x) = max\{1, \log|x|\},$$

$$2^* = \begin{cases} 2 & b_2 \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

3.2 The group $E(\mathbb{Q})$

We can define an addition operation to add two points on the curve together. The graph of an elliptic curve is such that any non-vertical line intersecting the curve at two points will intersect it at a third. The point at infinity O_E, can be thought of as a point so infinitely far up the y-axis that all vertical lines pass through it. We now have that a line through any two points (even if vertical) will pass through a third on the curve. Intersections with the curve are counted with multiplicity.

Example: Figure 3.4 shows the curve $y^2 = x^3 - 10x + 1$. Here a line has been drawn through the points $(-3, -2)$ and $(0, 1)$, and the line intersects the curve again at $(4, 5)$.

We can find the co-ordinates of this third point algebraically, given the co-ordinates of the first two points.

Lemma 3.2.1. *Let E be defined by equation (3.1), with $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E(\mathbb{Q})$. Then*

$$-P_1 = (x_1, -y_1 - a_1x_1 - a_3). \tag{3.7}$$

If $P_1 = -P_2$, then $P_3 = P_1 + P_2 = O_E$.

Otherwise $P_3 = P_1 + P_2 = (x_3, y_3)$, with

$$\begin{aligned} x_3 &= \lambda^2 + a_1\lambda - a_2 - x_1 - x_2 \\ y_3 &= -(\lambda + a_1)x_3 - \nu - a_3, \end{aligned} \tag{3.8}$$

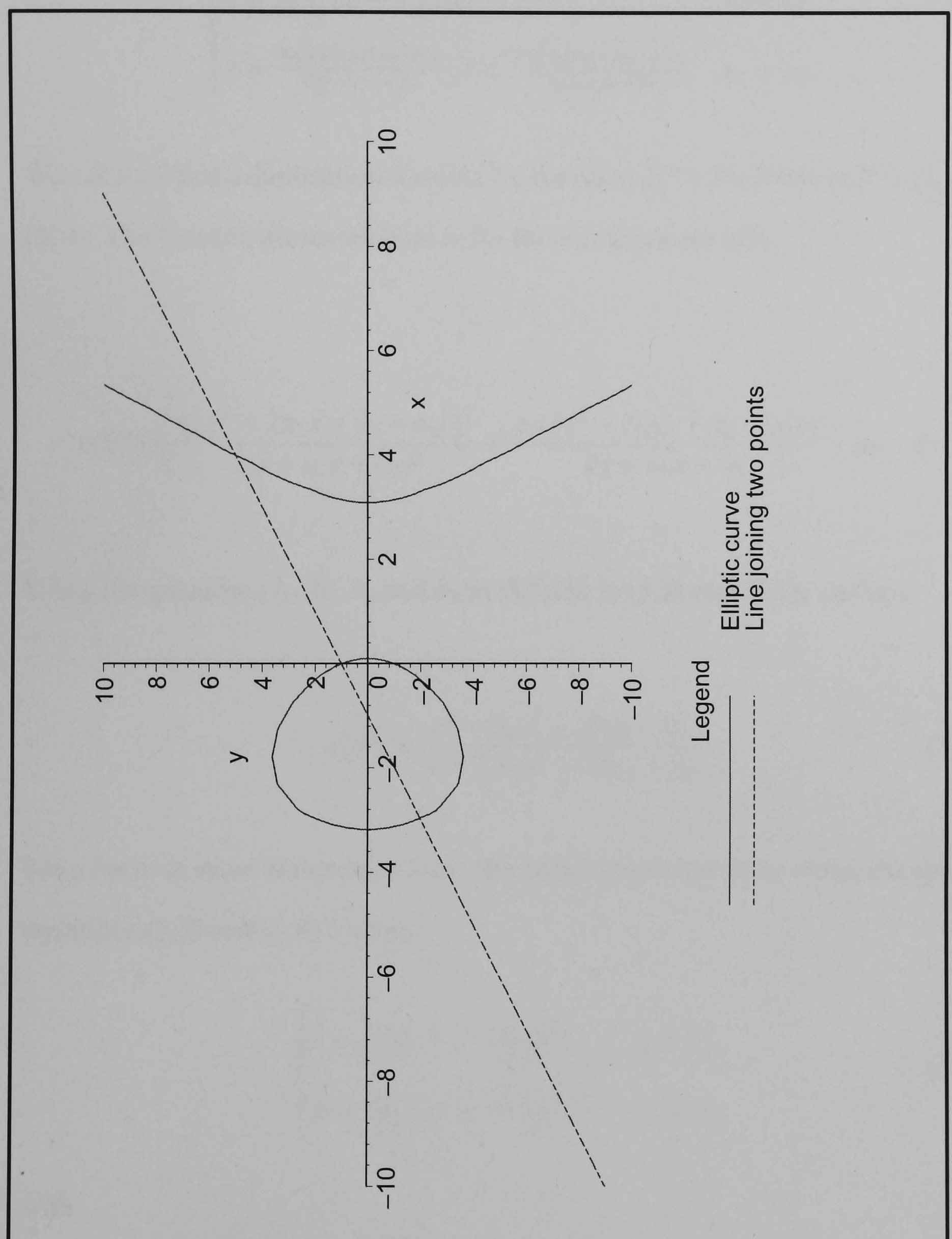

Figure 3.4: A straight line intersects an elliptic curve at three points

where

$$\begin{cases} \lambda = \frac{y_2-y_1}{x_2-x_1}, \nu = \frac{y_1x_2-y_2x_1}{x_2-x_1} & x_1 \neq x_2 \\ \lambda = \frac{3x_1^2+2a_2x_1+a_4-a_1y_1}{2y_1+a_1x_1+a_3}, \nu = \frac{-x_1^3+a_4x_1+2a_6-a_3y_1}{2y_1+a_1x_1+a_3} & x_1 = x_2. \end{cases} \tag{3.9}$$

We can now find a duplication formula for the point $2P = P+P$ where $P = (x, y) \neq (0, 0)$. The formula presented here is for the x-component only.

$$x(2P) = \frac{(3x^2 + 2a_2x + a_4 - a_1y)^2}{(2y + a_1x + a_3)^2} + \frac{a_1(3x^2 + 2a_2x + a_4 - a_1y)}{2y + a_1x + a_3} - a_2 - 2x.$$

Using the quantities b_2, b_4, b_6 and b_8 as defined in (3.3) and (3.5), we have

$$x(2P) = \frac{x^4 - b_4x^2 - 2b_6x - b_8}{4x^3 + b_2x^2 + 2b_4x + b_6}. \tag{3.10}$$

For a curve in short Weierstrass form, the addition process is the same, except that equations (3.9) and (3.8) become

$$\begin{cases} \lambda = \frac{y_2-y_1}{x_2-x_1}, \nu = \frac{y_1x_2-y_2x_1}{x_2-x_1} & x_1 \neq x_2 \\ \lambda = \frac{3x_1^2+A}{2y_1}, \nu = \frac{-x_1^3+Ax_1+2B}{2y_1} & x_1 = x_2, \end{cases} \tag{3.11}$$

with

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= -\lambda x_3 - \nu. \end{aligned} \tag{3.12}$$

The point $(x_1, y_1) + (x_2, y_2)$ is not the third point of intersection of the straight line with the curve, but its reflection in the x-axis.

Example: Figure 3.5 shows the addition of the points $(-3, -2)$ and $(0, 1)$ to obtain the point $(4, -5)$. The straight line drawn through the points $(-3, -2)$ and $(0, 1)$ intersects the curve again at $(4, 5)$. The reflection in the x-axis of this point is $(4, -5)$.

Clearly, if the points (x_1, y_1) and (x_2, y_2) are rational, then the point (x_3, y_3) will also be rational.

The Mordell-Weil theorem states that the group of rational points $E(\mathbb{Q})$ on the elliptic curve E is finitely generated. The proof relies on the following four lemmas, as shown in [43].

Lemma 3.2.2. *For every finite rational number M, the set $\{P \in E(\mathbb{Q}) : h(P) < M\}$ is finite.*

Proof. By definition 3.1.5, we have

$$H\left(\frac{m}{n}\right) = \max\left(|m|, |n|\right).$$

Since $|m|$ and $|n|$ are bounded, the result follows. □

Lemma 3.2.3. *Let $P_0 \in E(\mathbb{Q})$ be a fixed point on the curve E. Then there is a constant c_0, depending on P_0 such that*

$$h(P + P_0) \leq 2h(P) + c_0,$$

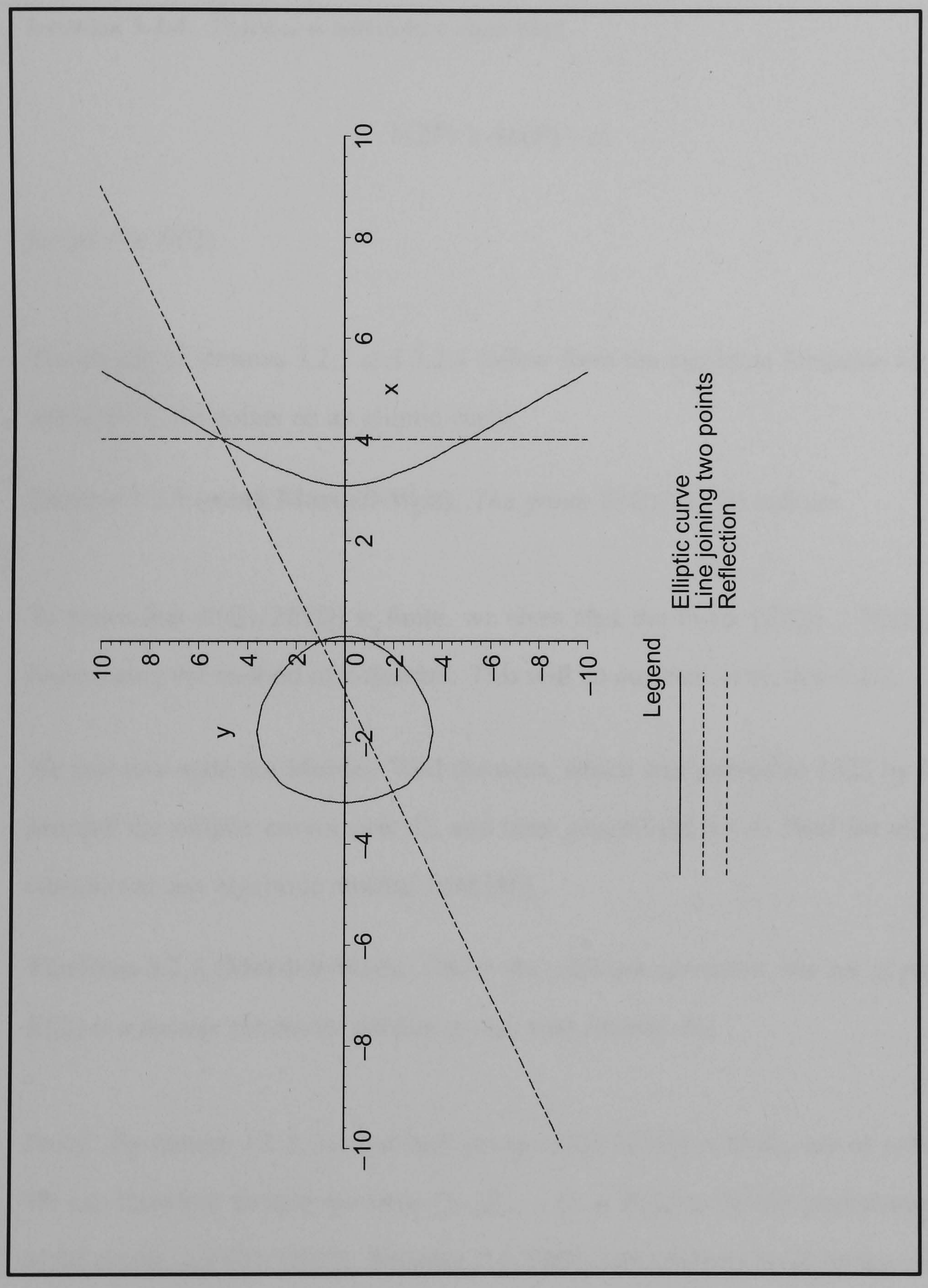

Figure 3.5: Addition of points on an elliptic curve

for all $P \in E(\mathbb{Q})$.

Lemma 3.2.4. *There is a constant c such that*

$$h(2P) \geq 4h(P) - c,$$

for all $P \in E(\mathbb{Q})$.

The proofs of lemmas 3.2.3 and 3.2.4 follow from the algebraic formulae for the addition of two points on an elliptic curve.

Lemma 3.2.5 (weak Mordell-Weil). *The group* $E(\mathbb{Q})/2E(\mathbb{Q})$ *is finite.*

To prove that $E(\mathbb{Q})/2E(\mathbb{Q})$ is finite, we show that the index $[E(\mathbb{Q}) : 2E(\mathbb{Q})]$ is finite, using the method of 2-descent. This will be outlined in section 3.2.2.

We can now state the Mordell-Weil theorem, which was proved in 1922 by L. J. Mordell for elliptic curves over $\mathbb{Q}$, and later generalised by A. Weil for elliptic curves over any algebraic number field [46].

Theorem 3.2.6 (Mordell-Weil). *Under the addition operation, the set of points* $E(\mathbb{Q})$ *is a finitely generated abelian group, with identity* O_E.

Proof. By lemma 3.2.5, the quotient group $E(\mathbb{Q})/2E(\mathbb{Q})$ is finite, say of order r. We can therefore choose elements $Q_1, Q_2, ..., Q_r \in E(\mathbb{Q})$ to be the representatives of the cosets in $E(\mathbb{Q})/2E(\mathbb{Q})$. Suppose $P \in E(\mathbb{Q})$ is an arbitrary point on the curve.

Let

$$\begin{aligned} P &= 2P_1 + Q_{i_1}, \quad 1 \le i_1 \le r \\ P_1 &= 2P_2 + Q_{i_2}, \quad 1 \le i_2 \le r \\ &\vdots \\ P_{n-1} &= 2P_n + Q_{i_n}, \quad 1 \le i_n \le r. \end{aligned} \tag{3.13}$$

Hence

$$P = 2^n P_n + \sum_{j=1}^{n} 2^{j-1} Q_{i_j}. \tag{3.14}$$

For any j, we have

$$\begin{aligned} h\left(P_j\right) &\le \frac{1}{4}\left[h\left(2P_j\right) + c\right] && \text{from lemma 3.2.4} \\ &= \frac{1}{4}\left[h\left(P_{j-1} - Q_{i_j}\right) + c\right] && \text{by (3.13).} \end{aligned}$$

By lemma 3.2.3, we have

$$h\left(P_{j-1} - Q_{i_j}\right) \le 2h\left(P_{j-1}\right) + c_{i_j}$$

where the constant c_{i_j} depends on the point Q_{i_j}. Since $Q_{i_j} \in \{Q_1, Q_2, \ldots, Q_r\}$, we can take c_0 to be the maximum of these constants.

Then

$$h\left(P_j\right) \le \frac{1}{4}\left[2h\left(P_{j-1}\right) + c_0 + c\right].$$

So

$$\begin{aligned} h\left(P_n\right) &\le 2^{-n} h\left(P\right) + \left(c_0 + c\right) \sum_{k=2}^{n+1} 2^{-k} \\ &\le 2^{-n} h\left(P\right) + \frac{c_0 + c}{2}. \end{aligned}$$

On taking n to be sufficiently large, we have

$$h(P_n) \leq 1 + \frac{c_0 + c}{2}.$$

From equation (3.14), every point $P \in E(\mathbb{Q})$ is a linear combination of points in the set $\{Q_1, Q_2, ..., Q_r\} \cup \{Q \in E(\mathbb{Q}) : h(Q) \leq 1 + \frac{c_0+c}{2}\}$. By lemma 3.2.2, this must be a finite set, and it follows that $E(\mathbb{Q})$ is finitely generated. □

Definition 3.2.7. The *order* of a point P in $E(\mathbb{Q})$ is the smallest number m such that $mP = O_E$.

Definition 3.2.8. A *torsion point* is a point of finite order.

Definition 3.2.9. The *rank* of the curve is the size of the basis for the subgroup of points of infinite order. If this is non-zero, then there are an infinite number of rational points on the curve.

We can now state the structure of the group of rational points $E(\mathbb{Q})$.

Theorem 3.2.10 (Mordell). *If E is an elliptic curve over $\mathbb{Q}$, then $E(\mathbb{Q})$ is a finitely generated abelian group, with*

$$E(\mathbb{Q}) \simeq E(\mathbb{Q})_{tors} \oplus \mathbb{Z}^r,$$

where r is the rank of the curve and $E(\mathbb{Q})_{tors}$ is the torsion subgroup.

So to compute $E(\mathbb{Q})$, we need to find the torsion subgroup $E(\mathbb{Q})_{tors}$, calculate the rank r, and find a set of r generators of the free abelian part $\mathbb{Z}^r$.

3.2.1 Computing the Torsion Subgroup

Finding the torsion subgroup is a relatively simple task, due to the following theorem of Nagell [36] and Lutz [27].

Theorem 3.2.11 (Nagell-Lutz). *Let* (x, y) *be a rational point of finite order on an elliptic curve* E *in short Weierstrass form (3.2) with discriminant* Δ*. Then* $x, y \in \mathbb{Z}$ *and either* $y = 0$*, in which case the point* (x, y) *has order* 2*, or* $y^2 \mid \Delta$*.*

So to find the torsion points, we calculate the discriminant Δ, and find all possible divisors y^2. We then solve the resulting cubic equations in x, and check that there is an integral solution. Having found the integral points P, we successively compute multiples mP until either $mP = O_E$ (in which case P has order m), or mP is not integral (in which case P has infinite order). Pseudocode for this algorithm is given by Cremona [12].

Example: Consider the curve $y^2 = x^3 - 36x$. The curve has discriminant $\Delta = 2985984$, so possible values of y are given by $y = 2^{\alpha}.3^{\beta}$, where $\alpha \in \{0, 1, 2, 3, 4, 5, 6\}$ and $\beta \in \{0, 1, 2, 3\}$. Solving $y^2 = x^3 - 36x$ leads to the integral solutions $(x, y) = (-3, 9), (12, 36), (-2, 8), (18, 72), (0, 0), (6, 0), (-6, 0)$. The first four points P all have a non-integral $2P$ and so are not of finite order. The other three points all have a zero y-coordinate, and so are of order 2.

The number of possible torsion points is therefore finite, and Mazur [30, 31] proved that the torsion subgroup is isomorphic to one of only 15 finite abelian groups.

Theorem 3.2.12 (Mazur). *The torsion subgroup* $E(\mathbb{Q})_{tors}$ *is one of the following 15 groups:*

$$\mathbb{Z}/N\mathbb{Z}, \quad 1 \leq N \leq 10 \text{ or } N = 12$$
$$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2N\mathbb{Z}, \quad 1 \leq N \leq 4.$$

3.2.2 Calculating the Rank

Bounds on the rank of an elliptic curve and a set of coset representatives for $2E(\mathbb{Q})$ in $E(\mathbb{Q})$, can be found using a 2-descent process. The process also gives a proof of the weak Mordell-Weil theorem (lemma 3.2.5).

The aim is to show that the image of the multiplication-by-two map $[2] : E(\mathbb{Q}) \rightarrow E(\mathbb{Q})$ has finite index. This can be achieved by demonstrating a one-to-one map from $E(\mathbb{Q})/2E(\mathbb{Q})$ into some finite group. To do this, we show that $2E(\mathbb{Q})$ is the kernel of a map from $E(\mathbb{Q})$ into some finite group.

Computationally, we first find a finite set of auxiliary curves, H, called *homogeneous spaces*, each isomorphic to the elliptic curve over some extension field (although not necessarily over $\mathbb{Q}$ itself). We then determine whether each space contains a rational point. The points on these homogeneous spaces correspond to rational points on the elliptic curve by a polynomial mapping $H \rightarrow E$, and the set of points obtained cover the cosets of $2E(\mathbb{Q})$ in $E(\mathbb{Q})$.

Since the maps $H \rightarrow E$ have degree 2 or 4, the points on the homogeneous spaces have smaller height, and thus should be easier to find. The points found can then be mapped back onto $E(\mathbb{Q})$.

2-descent via 2-isogeny

If $E(\mathbb{Q})$ contains a rational torsion point of order 2, we can use the method of 2-descent via 2-isogeny.

We use the isogeny

$$\phi : E(\mathbb{Q}) \to E'(\mathbb{Q}),$$

and the dual isogeny

$$\phi' : E'(\mathbb{Q}) \to E(\mathbb{Q}).$$

These maps split the multiplication-by-two map $[2] : E(\mathbb{Q}) \to E(\mathbb{Q})$ into two pieces, as shown in figure 3.6.

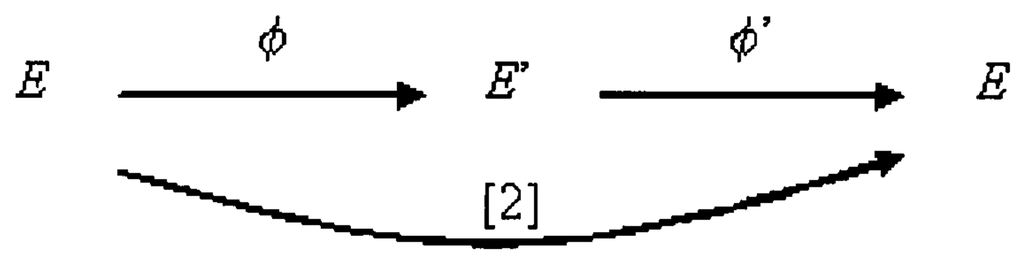

Figure 3.6: The multiplication-by-2 map

We then need to show that the image of each map has finite index. $E(\mathbb{Q})/\phi'(E'(\mathbb{Q}))$ and $E'(\mathbb{Q})/\phi(E(\mathbb{Q}))$ are each isomorphic to a subgroup of $\mathbb{Q}^*/(\mathbb{Q}^*)^2$, the size of which can be found computationally. The method, as described in [12], is as follows.

Suppose E is an elliptic curve in long Weierstrass form (3.1), and the quantities b_2, b_4 and b_6 are given by (3.3).

Let x_0 be a root of the cubic $x^3 + b_2x^2 + 8b_4x + 16b_6$. Then E is equivalent to

$$y^2 = x(x^2 + cx + d),$$

where

$$c = 3x_0 + b_2$$

$$d = (c + b_2)x_0 + 8b_4.$$

Note that if $a_1 = a_3 = 0$, we can take x_0 to be a root of $x^3 + a_2x^2 + a_4x + a_6$, and we can set $c = 3x_0 + a_2$, $d = (c + a_2)x_0 + a_4$.

Let

$$E' : y^2 = x(x^2 + c'x + d'),$$

where

$$c' = -2c$$

$$d' = c^2 - 4d.$$

Let the point at infinity on the curve E' be $0_{E'}$. Then the mappings ϕ and ϕ' are given by

$$\phi : E(\mathbb{Q}) \to E'(\mathbb{Q})$$

$$P \to \begin{cases} \left(\frac{y^2}{x^2}, \frac{y(x^2-d)}{x^2}\right) & P = (x,y) \neq (0,0), 0_E \\ 0_{E'} & P = (0,0), 0_E \end{cases} \tag{3.15}$$

and

$$\phi' : E'(\mathbb{Q}) \to E(\mathbb{Q})$$

$$P \to \begin{cases} \left(\frac{y^2}{4x^2}, \frac{y(x^2-d')}{8x^2}\right) & P = (x,y) \neq (0,0), 0_{E'} \\ 0_E & P = (0,0), o_{E'} \end{cases} \tag{3.16}$$

These maps are homomorphisms, with $\ker(\phi) = \{0_E, (0,0)\}$ and $\ker(\phi') = \{0_{E'}, (0,0)\}$.

Let $P = (x,y) \neq (0,0), 0_E$ be a rational point on the curve E. Then

$$\begin{aligned} \phi'(\phi(x,y)) &= \phi'\left(\frac{y^2}{x^2}, \frac{y(x^2-d)}{x^2}\right) \\ &= \left(\frac{(x^2-d)^2}{4y^2}, \frac{x^2(x^2-d)\left(\frac{y^4}{x^2}-d\right)}{8y^3}\right) \\ &= \left(\frac{x^4-2dx^2+d^2}{4y^2}, \frac{y^4-dx^4-d\frac{y^4}{x^2}+d^2x^2}{8y^3}\right) \\ &= [2](x,y). \end{aligned}$$

Similarly, the composition $\phi(\phi'(x,y))$ takes the point (x,y) to its double on the curve E'.

We now construct the homogeneous spaces H such that the diagram in figure 3.7 commutes.

For each square-free divisor d_1 of d, the homogeneous space

$$H(d_1, c) : v^2 = d_1u^4 + cu^2 + \frac{d}{d_1} \tag{3.17}$$

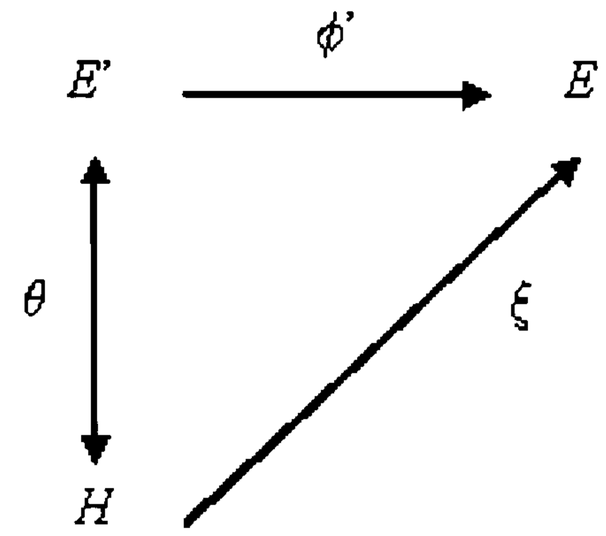

Figure 3.7: 2-descent via 2-isogeny

is examined for solutions. Methods for determining whether a given quartic has local and global solutions are given in [12].

Let $n_1(d,c)$ be the number of factors d_1 of d for which $H(d_1,c)$ has a rational point, and $n_2(d,c)$ be the number of factors d_1 of d for which $H(d_1,c)$ has a point everywhere locally. We also define $n_1' = n_1(d',c')$ and $n_2' = n_2(d',c')$.

Each rational point $P = (u,v)$ on $H(d_1,c)$ maps to a point $\xi(P) = (d_1u^2, d_1uv)$ on E, and each point $P = (u,v)$ on $H(d_1',c')$ maps to a point

$$\phi' \circ \theta(P) = \left(\frac{v^2}{4u^2}, \frac{v\left(d_1'u^4 - \frac{d'}{d_1'}\right)}{8u^3} \right)$$

on E.

By adding these points, we obtain a set of n_1n_1' points on E which gives either a complete set of coset representatives for $E(\mathbb{Q})/2E(\mathbb{Q})$, or a set of points which covers each coset exactly twice.

$E(\mathbb{Q})/\phi'(E'(\mathbb{Q}))$ is isomorphic to a subgroup of $\mathbb{Q}^*/(\mathbb{Q}^*)^2$ generated by factors d_1 for which $H(d_1, c)$ has a rational point. So

$$|E(\mathbb{Q})/\phi'(E'(\mathbb{Q}))| = n_1 \quad = 2^{e_1}$$
$$|E'(\mathbb{Q})/\phi(E(\mathbb{Q}))| = n_1' \quad = 2^{e_1'},$$

and it can be shown that

$$\text{rank}(E(\mathbb{Q})) = \text{rank}(E'(\mathbb{Q})) = e_1 + e_1' - 2.$$

This method will not always give the rank exactly, as some of the homogeneous spaces H may fail the Hasse principle (see section 2.3). If all of the spaces having points everywhere locally have a rational point, then $n_1 = n_2$. However, if the Hasse principle fails, then having determined that the space has a point everywhere locally, we might not be able to find a rational point. So the number of spaces having a rational point might be underestimated. In this case, all we know is that $n_1 \leq n_2$, and a bound on the rank is given. The extent to which this failure occurs is given by the size of the Tate-Shafarevich groups $Ш(E/\mathbb{Q})$ and $Ш(E'/\mathbb{Q})$ for the elliptic curves E and E' over $\mathbb{Q}$.

It is also possible that our search for a rational point on a homogeneous space $H(d_1, c)$ is inconclusive, due to the search bounds being insufficiently large. At this stage, we cannot determine whether failure to find a global rational point is due to a non-trivial element of the Tate-Shafarevich group, or whether the rational point is too large (in terms of height) to be found by the search. A second descent

distinguishes between these two cases.

General 2-descent

The method of 2-descent via isogeny outlined above depends on the existence of at least one rational 2-torion point. If no such point exists, we must use the general 2-descent method.

The general method was described by Birch and Swinnerton-Dyer [3], and has since been improved by Cremona [14].

This method uses homogeneous spaces H of the form

$$H: \quad y^2 = g(x) = ax^4 + bx^3 + cx^2 + dx + e \tag{3.18}$$

with $a, b, c, d, e \in \mathbb{Q}$, and

$$I = 12ae - 3bd + c^2,$$

$$J = 72ace + 9bcd - 27ad^2 - 27eb^2 - 2c^3.$$

The quantities I and J are related to the c_4 and c_6 invariants of E. Each of the curves H is a 2-covering of the curve

$$E_{I,J}: Y^2 = F(X) = X^3 - 27IX - 27J, \tag{3.19}$$

as shown in figure 3.8.

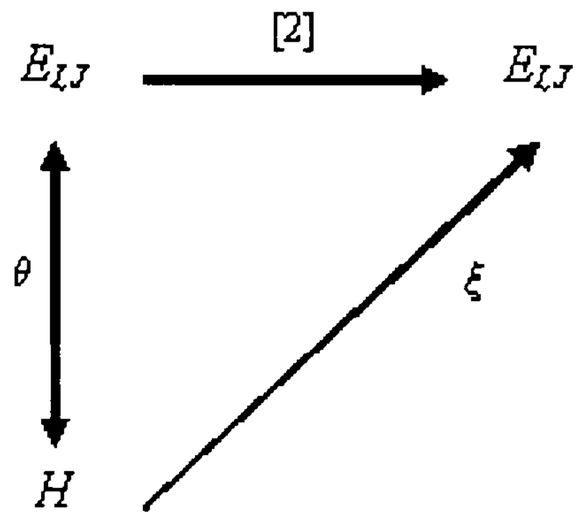

Figure 3.8: General 2-descent

Here, the map θ is a birational map from H to its jacobian (which is $E_{I,J}$).

The process, as described in [12], is as follows.

Step 1

Determine the invariant pair (I, J).

Let $I = c_4$ and $J = 2c_6$, where c_4 and c_6 are as defined in (3.4). If $2^4 \mid I$ and $2^6 \mid J$, we replace (I, J) by $(2^{-4}I, s^{-6}J)$, and use both this pair and the pair $(16I, 64J)$ in the remaining steps, unless $4 \mid I$, $8 \mid J$ and $16 \mid (2I + J)$ in which case we use only the pair (I, J).

Step 2

Find all quartics $g(x)$ *with the given pair* (I, J).

The quartics can be categorised as those with no real roots (type 1), those with four real roots (type 2), and those with two real roots (type 3).

Bounds on the coefficients a, b and c are first determined. The auxiliary cubic equation

$$\phi^3 - 3I\phi + J = 0 \tag{3.20}$$

will have one real root (type 3) or three real roots (types 1 and 2).

For the quartics with no real roots (type 1), order the roots of equation (3.20) as $\phi_1 > \phi_2 > \phi_3$ and set $K = \frac{4I-\phi_1^2}{3}$. Then

$$0 < a \leq \frac{K + K^{1/2}\phi_1}{3K^{1/2} + \phi_1 + 2\phi_2},$$

$$-2a < b \leq 2a,$$

$$\frac{4a\phi_2 + 3b^2}{8a} \leq c \leq \frac{4a\phi_1 + 3b^2}{8a}.$$

For the quartics with four real roots (type 2), the bounds depend on the sign of a.

If $a > 0$, order the roots of equation (3.20) as $\phi_1 > \phi_2 > \phi_3$. Then

$$0 < a \leq \frac{I - \phi_2^2}{3(\phi_2 - \phi_3)},$$

$$-2a < b \leq 2a,$$

$$\frac{4a\phi_2 - \frac{4}{3}(I - \phi_2^2) + 3b^2}{8a} \leq c \leq \frac{4a\phi_3 + 3b^2}{8a}.$$

If $a < 0$, order the roots of equation (3.20) as $\phi_1 < \phi_2 < \phi_3$. Then

$$0 < -a \leq \frac{I - \phi_2^2}{3(\phi_3 - \phi_2)},$$

$$-2 \mid a \mid < b \leq 2 \mid a \mid,$$

$$\frac{4a\phi_2 - \frac{4}{3}(I - \phi_2^2) + 3b^2}{8a} \geq c \geq \frac{4a\phi_3 + 3b^2}{8a}.$$

For the quartics with two real roots (type 3), equation (3.20) has a unique real root ϕ. Then

$$\frac{1}{3}\phi - \sqrt{\frac{4}{27}(\phi^2 - I)} \le a \le \frac{1}{3}\phi + \sqrt{\frac{4}{27}(\phi^2 - I)},$$

$$-2 \mid a \mid < b \le 2 \mid a \mid,$$

$$\frac{9a^2 - 2a\phi + \frac{1}{3}(4I - \phi^2) + 3b^2}{8 \mid a \mid} \le c.\mathrm{sign}(a) \le \frac{4a\phi + 3b^2}{8 \mid a \mid}.$$

Associated to each of the quartics g are the two *covariants*,

$$\begin{aligned} g_4(X, Y) = \quad & (3b^2 - 8ac)X^4 + 4(bc - 6ad)X^3Y + 2(2c^2 - 24ae - 3bd)X^2Y^2 \\ & + 4(cd - 6bc)XY^3 + (3d^2 - 8ce)Y^4 \end{aligned}$$

$$\begin{aligned} g_6(X, Y) = \quad & (b^3 + 8a^2d - 4abc)X^6 + 2(16a^2e + 2abd - 4ac^2 + b^2c)X^5Y \\ & + 5(8abe + b^2d - 4acd)X^4Y^2 + 20(b^2e - ad^2)X^3Y^3 \\ & - 5(8ade + bd^2 - 4bce)X^2Y^4 - 2(16ae^2 + 2bde - 4c^2e + cd^2)XY^5 \\ & - (d^3 + 8be^2 - 4cde)Y^6. \end{aligned}$$

Taking

$$g(X, Y) = aX^4 + bX^3Y + cX^2Y^2 + dXY^3 + e$$

so that $H : y^2 = g(x, 1) = ax^4 + bx^3 + cx^2 + dx + e$, we find that the polynomials $g_4(X, Y)$, $g_6(X, Y)$ and $g(X, Y)$ must satisfy the syzygy

$$27g_6^2 = g_4^3 - 48Ig^2g_4 - 64Jg^3. \tag{3.21}$$

Next, we define the *semivariants*,

$$p = g_4(1,0) = 3b^2 - 8ac$$
$$r = g_6(1,0) = b^3 + 8a^2d - 4abc. \tag{3.22}$$

On substituting $(X, Y) = (1, 0)$ into (3.21), we obtain the semivariant syzygy

$$27r^2 = p^3 - 48Ia^2p - 64Ja^3. \tag{3.23}$$

Therefore, variables a, b and c must satisfy equation (3.23).

Then d and e are given by

$$d = \frac{r - b^3 + 4abc}{8a^2},$$
$$e = \frac{I + 3bd - c^2}{12a}.$$

Step 3

Test for equivalence of quartics.

Two quartics $g_1(x)$ and $g_2(x)$ are equivalent if

$$g_2(x) = \mu^2(\gamma x + \delta)^4 g_1\left(\frac{\alpha x + \beta}{\gamma x + \delta}\right)$$

for some $\alpha, \beta, \gamma, \delta, \mu \in \mathbb{Q}$ with $(\alpha\delta - \beta\gamma)$ and μ both non-zero. Any quartics found which are equivalent to an earlier quartic are discarded.

Step 4

Test for local and global solubility.

The number of equivalence classes of quartics $g(x)$ which are everywhere locally soluble is finite. Let the number of quartics which are found to have a rational point be n_1 and the number of quartics found to have a point everywhere locally be n_2. Then we should have $n_i = 2^{e_i}$ for $i = 1, 2$.

Step 5

Compute the rank.

The rank of $E(\mathbb{Q})$ is given by

$$e_1 - 1 \leq \text{rank}\,(E(\mathbb{Q})) \leq e_2 - 1.$$

If we have found a (global) rational point on all n_2 quartics known to be everywhere locally soluble, then $n_1 = n_2$ and we can calculate the rank exactly.

If n_1 is not found to be a power of 2, then we must have failed to find a rational point on a quartic where such a point exists. In this case, we may replace n_1 by the next highest power of 2. This failure could be due to a non-trivial element of the Tate-Shafarevich group $Ш(E/\mathbb{Q})$, or the height of the rational point is too large to be found by the search. A second descent would be needed in order to determine whether a rational point exists.

Step 6

Recover points on $E(\mathbb{Q})$.

Each quartic $g(x)$ for which a rational point P is found leads to a point $\xi(P)$ on the curve $E_{I,J}$ via the map

$$\xi : H(\mathbb{Q}) \to E_{I,J}(\mathbb{Q})$$

$$(x,y) \to \begin{cases} \left(\frac{3g_4(x,1)}{(2y)^2}, \frac{27g_6(x,1)}{(2y)^3}\right) & (x,y) \neq 0_H \\ \left(\frac{3p}{4a}, \frac{\pm 27r}{(4a)^{3/2}}\right) & (x,y) = 0_H \end{cases}$$

where 0_H is the point at infinity on H.

This is a degree 4 map, taking rational points on H satisfying $y^2 = g(x,1)$ to rational points on $E_{I,J}$. These points are then mapped onto the original curve E.

If this map is applied to the rational points found on each of the n_1 quartics from step 5, we obtain a complete set of coset representatives for $2E(\mathbb{Q})$ in $E(\mathbb{Q})$. In the case where we failed to find some of the rational points (so that n_1 is not a power of 2), we will still have generators for $E(\mathbb{Q})/2E(\mathbb{Q})$, and can use these to determine the remaining coset representatives.

Higher descents

For some elliptic curves, the first descent is inconclusive. We may obtain homogeneous spaces which are everywhere locally soluble but for which no rational point is found. The failure will be due either to a non-trivial element of the Tate-Shafarevich group or to the search bounds being insufficiently large. A second

descent is used to determine which of these two cases is true, by either providing a large rational point on the homogeneous space H as the image of a smaller point on a descendant curve D of H, or by proving that no such descendant exists.

In the case of 2-descent via 2-isogeny, a homogeneous space H is a ϕ'-covering of E, as shown in figure 3.7. A second descent extends the ϕ'-covering to a 2-covering D so that the diagram in figure 3.9 commutes. The vertical maps θ and θ_2 are isomorphisms defined over $\bar{\mathbb{Q}}$, while the remaining maps are all of degree 2 and defined over $\mathbb{Q}$.

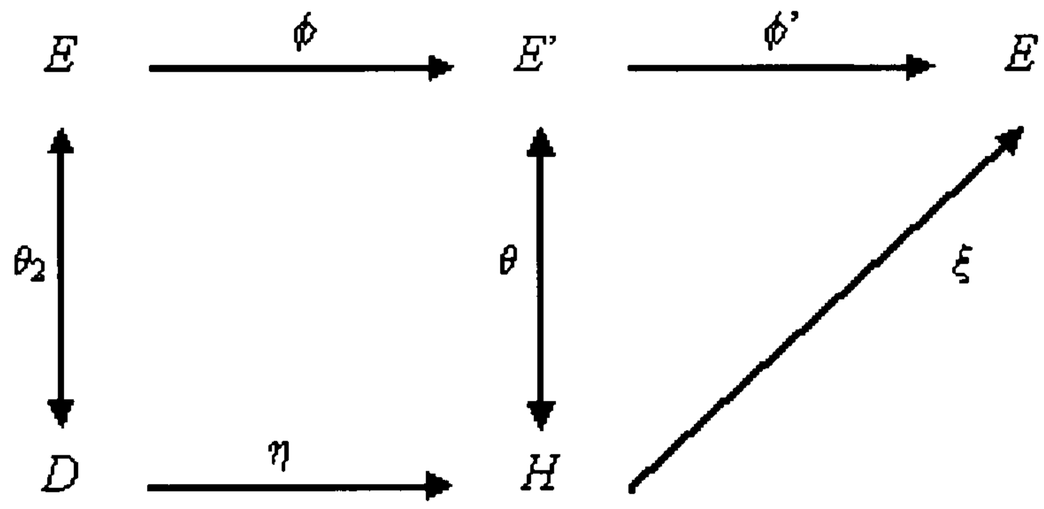

Figure 3.9: The second descent in the 2-isogeny case

The homogeneous space D is called a *descendant* of H. If a rational point P is found on one of the descendants of H, then $\eta(P)$ is a rational point on H, and hence $\xi \circ \eta(P)$ is a rational point on E. Since the curves D are 2-coverings of E, the rational points on D should have smaller height than those on E by a factor of 2.

If it is found that a homogeneous space H has no everywhere locally soluble descendants, then H has no rational points and hence represents a non-trivial element of the Tate-Shafarevich group $Ш(E'/\mathbb{Q})$. If H has descendants which are every-

where locally soluble but no rational point can be found, then H represents an element of $Ш(E'/\mathbb{Q})[2]$.

Second (and higher) descents have been used for various families of curves with rational 2-torsion (see, for example, [37, 8, 33, 48, 40]). The method of Bremner and Cassels [8] is presented in chapter 4. The general method, as described in [13], was first used by Birch and Swinnerton-Dyer [4].

Suppose that $H(d_1, c)$ is an everywhere locally soluble curve given by

$$H(d_1, c): \quad v^2 = d_1u^4 + cu^2 + \frac{d}{d_1}, \tag{3.24}$$

for which we wish to find a rational solution (u, v).

Step 1

Solve the conic

Let

$$u^2 = \frac{X}{Z}, \qquad v = \frac{Y}{Z}. \tag{3.25}$$

- On substituting (3.25) into (3.24), we obtain the conic

$$H_0(d_1, c): \quad Y^2 = d_1X + cXZ + \frac{d}{d_1}Z^2. \tag{3.26}$$

Rational points on $H(d_1, c)$ correspond to rational points on $H_0(d_1, c)$ with $\frac{X}{Z}$ a

square. Since H is everywhere locally soluble, H_0 is also everywhere locally soluble, and so H_0 has a rational point $P_0 = (X_0 : Y_0 : Z_0) \in \mathbb{P}^2(\mathbb{Q})$ by the Hasse principle (see section 2.3). We first search for a solution $(X_0 : Y_0 : Z_0)$.

Step 2

Parameterise the conic

Given a solution P_0 of the conic (3.26), a parameterisation of all solutions is given by

$$\begin{aligned} X &= q_1(\alpha,\beta), \\ Y &= q_2(\alpha,\beta), \\ Z &= q_3(\alpha,\beta), \end{aligned} \tag{3.27}$$

where the q_i are quadratic polynomials with integer coefficients which determine a birational map

$$\begin{aligned} \zeta : \quad & \mathbb{P}^1/(\mathbb{Q}) \to H_0(\mathbb{Q}) \\ & (\alpha : \beta) \to (q_1(\alpha,\beta), q_2(\alpha,\beta), q_3(\alpha,\beta)) . \end{aligned}$$

The parameterisation can be chosen such that q_1 and q_3 have discriminants $16\frac{d}{d_1}$ and $16d_1$ respectively, with resultant $\mathrm{Res}_u\,(q_1(u, 1), q_3(u, 1)) = 16d'$.

Step 3

Parameterise the set of descendants

If $\frac{X}{Z} = \frac{q_1(\alpha,\beta)}{q_3(\alpha,\beta)}$ is a square, then $\zeta(\alpha,\beta)$ gives a rational point on H. Therefore, we

require an integer solution $(\alpha, \beta, \gamma, \delta)$ to the equations

$$q_1(\alpha, \beta) = d_3\gamma^2, \tag{3.28}$$
$$q_3(\alpha, \beta) = d_3\delta^2,$$

where d_3 is a square-free divisor of $16d'$.

The finite set of d_3 for which equations (3.28) are separately soluble are the integers which parameterise the first descent curves $H(d_3, c') : v^2 = d_3u^4 + c'u^2 + \frac{d}{d_3}$ for the curve E'. These integers d_3 form a group G modulo squares. As part of the first descent, we have already found a subgroup G_1 of the d_3 for which $H(d_3, c')$ is everywhere locally soluble, and there will be n_2' of them. This subgroup is either empty, or forms a complete coset of G_1 in G. We do not need to test all values of d_3, but only one in each G_1 coset. Each value of d_3 to be tested produces a descendant curve D.

The remaining steps are performed for each value of d_3.

Step 4

Construct the descendants

We first ensure that equations (3.28 are separately soluble, and discard any value of d_3 for which either equation has no solution.

Assuming both equations are soluble, we find a solution to the first equation, and

hence parameterise the solutions

$$\begin{aligned} \alpha &= Q_1(\lambda,\mu), \\ \beta &= Q_3(\lambda,\mu), \\ \gamma &= Q_2(\lambda,\mu), \end{aligned} \tag{3.29}$$

where the Q_i are quadratics with integer coefficients such that

$$q_1\left(Q_1(\lambda,\mu), Q_3(\lambda,\mu)\right) = d_3 Q_2(\lambda,\mu).$$

Substituting (3.29) into (3.28) gives the descendant curve

$$D: \quad g(\lambda,\mu) = d_3\delta^2, \tag{3.30}$$

where $g(\lambda,\mu)$ is the quartic $q_3\left(Q_1(\lambda,\mu), Q_3(\lambda,\mu)\right)$.

Step 5

Search for rational points on the descendants

We test for local solubility using the algorithm given in [12].

Suppose that D is everywhere locally soluble. A sieve assisted search is used to attempt to find a global rational point on D. If a rational point (λ,μ,δ) is found on D, it maps to a rational point (u,v) on H. Equations (3.25), (3.29) and (3.27)

provide this map explicitly, since

$$
\begin{aligned}
(u^2, v) &= \left(\frac{X}{Z}, \frac{Y}{Z}\right) \\
&= \left(\frac{q_1(\alpha,\beta)}{q_3(\alpha,\beta)}, \frac{q_2(\alpha,\beta)}{q_3(\alpha,\beta)}\right) \\
&= \left(\frac{d_3\gamma^2}{d_3\delta^2}, \frac{q_2(\alpha,\beta)}{d_3\delta^2}\right) \\
&= \left(\frac{\gamma^2}{\delta^2}, \frac{q_2(\alpha,\beta)}{d_3\delta^2}\right),
\end{aligned}
$$

so that

$$
\begin{aligned}
(u, v) &= \left(\frac{\gamma}{\delta}, \frac{q_2(\alpha,\beta)}{d_3\delta^2}\right) \\
&= \left(\frac{Q_2(\lambda,\mu)}{\delta}, \frac{q_2\left(Q_1(\lambda,\mu), Q_3(\lambda,\mu)\right)}{d_3\delta^2}\right).
\end{aligned}
$$

If no rational point is found, we test each of the elements of the d_3G_1 coset in turn. If no rational point is found with any of the coset elements, we either increase the search bound or attempt a 3rd descent. If none of the G_1 cosets produce an everywhere locally soluble descendant, then H represents an element of the Tate-Shafarevich group.

This method has been implemented in both Cremona's `mwrank` [16] and Connell's `apecs` [11].

The second descent in the case where there is no rational 2-torsion is more complicated, and requires algebraic number theory. The method is described in [32], which follows from the work of Siksek's thesis [42]. In this case, the descendant curve D will be a 4-covering of $E_{I,J}$, as shown in figure 3.10. The descent has been implemented in Magma version 2.11 (2004) by Womack [53] and subsequently

improved by Watkins for version 2.12 (2005).

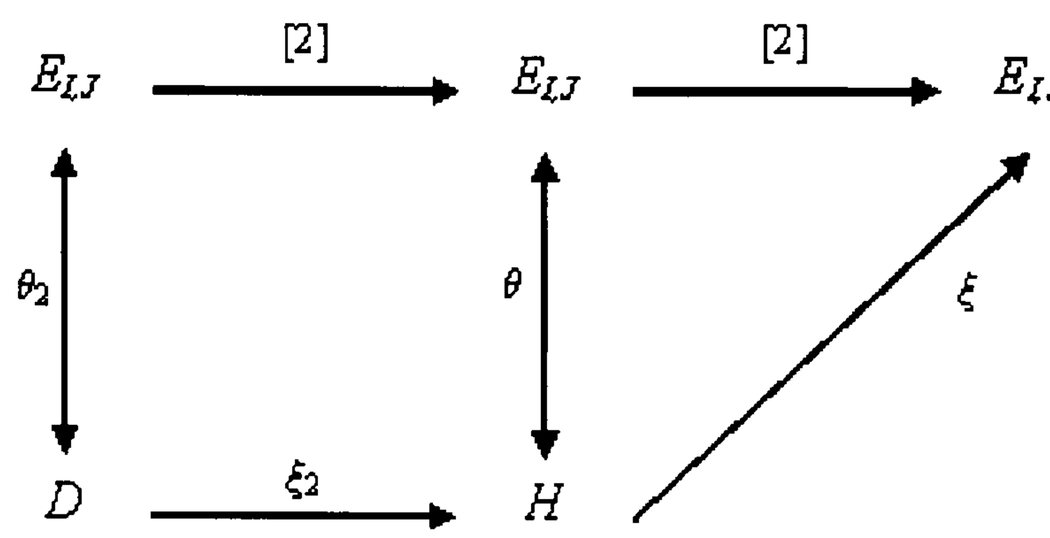

Figure 3.10: The second descent in the general case

Higher descents on elliptic curves will produce covering curves which are given as the intersection of two quadrics (a pair of simultaneous quaternary quadratic equations). Methods for solving the simultaneous equations representing the quadric intersection are presented in chapter 6.

3.2.3 Finding Generators

Having calculated the rank r of the elliptic curve E and a set of coset representatives for $E(\mathbb{Q})/2E(\mathbb{Q})$, finding the generators of the free abelian part $\mathbb{Z}^r$ is a two-stage process [12]. We first find all points P whose logarithmic height $h(P)$ is bounded by some constant N. Each point found is passed to the second stage of the algorithm, which determines whether the point is of infinite order, and whether it is independent of the points already found. By adjusting the value of N, a basis of r points is constructed.

This process uses the following definitions.

Definition 3.2.13. The *regulator*, $R(E(\mathbb{Q}))$, of $E(\mathbb{Q})$ modulo torsion is defined to be

$$R(E(\mathbb{Q})) = \det\left(\hat{h}(P_i, P_j)\right),$$

where the P_i form a basis for $E(\mathbb{Q})/E(\mathbb{Q})_{tors}$ and $\hat{h}(P_i, P_j)$ is the *Neron-Tate height pairing*,

$$\hat{h}(P, Q) = \frac{1}{2}\left(\hat{h}(P + Q) - \hat{h}(P) - \hat{h}(Q)\right).$$

We first determine a bound N for the maximum height of a set of generators, using the following lemma.

Lemma 3.2.14. *Let* $m > 0$ *be such that*

$$S = \{P \in E(\mathbb{Q}) : \hat{h}(P) \leq m\}$$

contains a complete set of coset representatives for $E(\mathbb{Q})/2E(\mathbb{Q})$. *Then* S *generates* $E(\mathbb{Q})$.

Having obtained a complete set of coset representatives for $E(\mathbb{Q})/2E(\mathbb{Q})$ during the descent process of section 3.2.2, we compute the canonical height of these points to give a value of m for which lemma 3.2.14 holds, then add the maximum difference between the logarithmic and canonical heights as given in lemma 3.1.6. This gives a value for the bound N to be used in step 1 of the algorithm.

Step 1

Let E be an elliptic curve defined by equation (3.2). First find the point $(x_0, y_0) \in E(\mathbb{R})$ with minimal x-value on E. Since an elliptic curve is symmetrical about the x-axis, we have $y_0 = 0$. This gives a lower bound for the possible x-values in $E(\mathbb{Q})$.

By theorem 3.1.4, a rational point $P \in E(\mathbb{Q})$ is of the form $P = \left(\frac{l}{n^2}, \frac{m}{n^3}\right)$ with $gcd(l, n) = 1$, so that $h(P) = max\left(\log|l|, \log n^2\right)$. Without loss of generality, we may take n to be positive. Then for $h(P) \leq N$, we require $\max\left(|l|, n^2\right) \leq e^N$. This means that $n \leq e^{\frac{N}{2}}$, and $-e^N \leq l \leq e^N$. However, we know that x_0 is the minimum possible x-value, so that $\frac{l}{n^2} \geq x_0 \implies l \geq n^2 x_0$. We therefore have that $\max\left(n^2 x_0, -e^N\right) \leq l \leq e^N$.

For each pair of integers (l, n), we attempt to solve the quadratic equation $m^2 = l^3 + Aln^4 + Bn^6$ for $m \in \mathbb{Z}$. Note that this process can be made more efficient by using a quadratic sieve [12].

In this manner, all rational points P with $h(P) \leq N$ are determined. As each point is found, it is passed to step 2 of the algorithm.

Step 2

This step will be performed whenever a new point P with $h(P) \leq N$ is found.

At a general stage, we will have a basis $\{P_1, P_2, \ldots, P_k\}$ of independent points

which are the generators for a subgroup of rank k, and we will have calculated the $k \times k$ height pairing matrix $M_k = [\hat{h}(P_i, P_j)]$ and $R_k = \det(M_k)$.

We first test to see if the new point P is of infinite order, and ignore it if it is found to be a torsion point. If the point is found to be of infinite order, we set $P_{k+1} = P$, and complete the height pairing matrix M_{k+1} by calculating $\hat{h}(P, P_i)$, $1 \leq i \leq k+1$. We then find the determinant R_{k+1} of this new matrix.

If R_{k+1} is non-zero, the point P_{k+1} is independent of the previous k points P_i and is added to the basis. M_k and R_k are replaced by M_{k+1} and R_{k+1} respectively, and the value of k is increased by 1.

If R_{k+1} is zero, then the point P_{k+1} is a linear combination of the previous points modulo torsion. Thus we can write

$$a_1P_1 + a_2P_2 + \ldots + a_kP_k + a_{k+1}P_{k+1} = 0 \text{ (modulo torsion)},$$

with $a_i \in \mathbb{Z}$. We must have $a_{k+1} \neq 0$, since the first k points are independent. If $a_{k+1} = \pm 1$, then the point P_{k+1} is ignored. If $a_{k+1} \neq \pm 1$ but $a_i = \pm 1$ for some $i \leq k$, then the point P_i is replaced by P_{k+1} in the basis. This new basis of k points generates a larger group by a finite index $|a_{k+1}|$.

We then return to step 1 and search for the next point P.

The value of the bound N used in step 1 can be increased until we have a basis containing r points. The subgroup generated by these points has finite index in $E(\mathbb{Q})$, and must be enlarged to give the whole of $E(\mathbb{Q})$.

3.3 The Birch and Swinnerton-Dyer Conjecture

Associated with each elliptic curve E, we can define a function $L(E, s)$ called the *L-function*. The definition relies on the number of points lying on the reduced curve $\tilde{E}(\mathbb{F}_p)$ for each prime p.

Let N_p be the number of points on $\tilde{E}(\mathbb{F}_p)$ including the point at infinity O_E, and let $A_p = 1 + p - N_p$.

Then we can define

$$L(E, s) = \prod_{p|\Delta} \left(1 - A_p p^{-s}\right)^{-1} \prod_{p\nmid\Delta} \left(1 - A_p p^{-s} + p^{1-2s}\right)^{-1}.$$

If p is a prime of bad reduction (so that $p \mid \Delta$), then

$$A_p = \begin{cases} 1 & \text{if } \tilde{E} \pmod{p} \text{ has a node with tangents whose slopes are rational over } \mathbb{F}_p \\ -1 & \text{if } \tilde{E} \pmod{p} \text{ has a node with tangents quadratic over } \mathbb{F}_p \\ 0 & \text{if } \tilde{E} \pmod{p} \text{ has a cusp.} \end{cases}$$

The Taylor series expansion of $L(E, s)$ around $s = 1$ is given by $L(E, s) = \sum_{n=1}^{\infty} a_n n^{-s}$.

The Birch and Swinnerton-Dyer conjecture states that this function has a zero at $s = 1$ whose order is equal to the rank of $E(\mathbb{Q})$, and that the leading coefficient in the Taylor series expansion can be written in terms of various quantities associated with E.

Following Silverman's book ([43]), let

E be an elliptic curve defined over $\mathbb{Q}$,

$Ш(E,\mathbb{Q})$ be the Tate-Shafarevich group of E over $\mathbb{Q}$,

$R(E(\mathbb{Q}))$ be the regulator of $E(\mathbb{Q})/E(\mathbb{Q})_{tors}$,

ω be the invariant differential $\frac{dx}{(2y+a_1x+a_3)}$,

$E_0(\mathbb{Q}_p)$ be the set of non-singular points modulo p,

Ω $= \int_{E(\mathbb{R})} |\omega|$, (either the real period, or twice the real period),

c_p $= \left|E(\mathbb{Q}_p)/E_0(\mathbb{Q}_p)\right|$.

Conjecture 3.3.1 (Birch and Swinnerton-Dyer). *(a)* $L(E,s)$ *has a zero at* $s = 1$ *of order equal to the rank of* $E(\mathbb{Q})$.

(b) Let r *be the rank of* $E(\mathbb{Q})$. *Then*

$$\lim_{s\to 1}(s-1)^{-r}L(E,s) = \Omega\,|Ш(E/\mathbb{Q})|\,2^r R(E(\mathbb{Q}))\,|E(\mathbb{Q})_{tors}|^{-2}\prod_p c_p$$

This would give an exact formula for $L^{(r)}(E,1) \in \mathbb{R}$,

$$L^{(r)}(E,1) = \alpha.\Omega.R(E(\mathbb{Q})), \tag{3.31}$$

where α is a non-zero rational number [22].

3.4 Heegner points

In the case of rank 1 elliptic curves, we can construct Heegner points in order to find rational points of infinite order on E. A Heegner point is a quadratic surd τ on the upper half plane $\mathcal{H}$. The point can be represented by a positive definite binary quadratic form (A, B, C), so that $A\tau^2+B\tau+C = 0$, with $A > 0$ and $\gcd(A, B, C) = 1$. The discriminant of τ is $D = B^2 - 4AC$. A Heegner point is of level N and discriminant D if $N \mid A$, $\gcd\left(\frac{A}{N}, B, CN\right) = 1$ and D is a square modulo $4N$.

The construction of Heegner points is described in [54]. For a fixed positive integer N, let $X_0(N)$ be the modular curve classifying pairs (E_1, E_2) of elliptic curves together with a cyclic isogeny $\alpha : E_1 \to E_2$ of degree N.

The set of complex points on $X_0(N)$ is a surface admitting the uniformisation

$$\eta : \mathcal{H}^*/\Gamma_0(N) \to X_0(N)(\mathbb{C}),$$

where $\mathcal{H}^* = \mathcal{H} \cup \mathbb{P}_1(\mathbb{Q})$ is the extended upper half plane and $\Gamma_0(N)$ is the subset of elements of $\mathrm{SL}_2(\mathbb{Z})$ whose reductions modulo N are upper triangular. This map sends $\tau \in \mathcal{H}$ to the point χ of $X_0(N)(\mathbb{C})$ associated to the pair of elliptic curves $\left(\frac{\mathbb{C}}{<\tau,1>}, \frac{\mathbb{C}}{<\tau,\frac{1}{N}>}\right)$ together with the related N-isogeny.

By the work of Wiles and others [52, 49, 9], there is a map $X_0(N) \to E$. This map can be broken into two pieces. We first map the point $\chi \in X_0(N)$ to the point z of

$\mathbb{C}/\Lambda$, where Λ is a certain lattice related to E, using

$$\varphi(\chi) = -\sum_{n=1}^{\infty} \frac{a_n}{n} q^n \in \mathbb{C}/\Lambda, \tag{3.32}$$

where the a_i are the coefficients of $L(E, s) = \sum_{n=1}^{\infty} a_n n^{-s}$ and $q = e^{2\pi i \chi}$.

The map $\eta_E : \mathbb{C}/\Lambda \to E(\mathbb{C})$ is then given by

$$\eta_E(z) = (\wp_\Lambda(z), \wp'_\Lambda(z)),$$

where $\wp_\Lambda$ is the Weierstrass parameterisation attached to Λ. Explicit formulae for Λ and $\wp_\Lambda$ can be found in [43].

The following practical method for calculating Heegner points on $\mathcal{H}/\Gamma_0(N)$ is described in [17].

Let K be an imaginary quadratic subfield of $\mathbb{C}$, and let $\mathfrak{O}$ be an order in K. Then $\mathfrak{O}$ is completely determined by its discriminant, D. We find the Heegner points on $\mathcal{H}/\Gamma_0(N)$ attached to the order $\mathfrak{O}$ of discriminant D, where D is coprime to N.

We first choose an integer s satisfying the congruence $s^2 \equiv D \pmod{4N}$. This gives a cyclic $\mathfrak{O}$-ideal $\mathfrak{n} = \left(N, \frac{s-\sqrt{D_0}}{2}\right)$ of norm N.

The images under η of the Heegner points attached to $\mathfrak{O}$ for which $\ker(E_1 \to E_2) = E_1[\mathfrak{n}]$ are in bijection with the $\mathrm{SL}_2(\mathbb{Z})$-equivalence classes of binary quadratic forms $Ax^2 + Bxy + Cy^2$ satisfying $B^2 - 4AC = D$, $N \mid A$, and $B \equiv s \pmod{2N}$.

Let the quadratic form associated to the Heegner point τ be (A', B', C'). Under this

bijection, any point on $\mathcal{H}/\Gamma_0(N)$ which maps via η to (A', B', C') is in the class of τ. We can take τ to be the unique root of $Ax^2 + Bx + C$.

To find the binary quadratic forms, we let $c = \frac{s^2-D}{4N}$ and construct the forms (a_iN, s, c_i) where c_i is a divisor of c and $a_i = \frac{c}{c_i}$. These are then reduced using the theory of binary quadratic forms (see, for example, [10]). The Heegner point representatives are then given by $\tau_i = \frac{-s+\sqrt{D}}{2a_iN}$. In this way, we compute a list of representatives $\tau_1, \ldots, \tau_h \in \mathcal{H}$ (where h is the class number of $\mathfrak{O}$). If this is not the case, we choose another value for s.

We then find $\varphi(\tau_i)$ for each representative, and sum them in $\mathbb{C}/\Lambda$ to give the complex number z. Note that we might not need to compute all of the $\varphi(\tau_i)$, since if $\varphi(\tau_{i_2})$ is the complex conjugate of $\varphi(\tau_{i_1})$ then $\varphi(\tau_{i_1}) + \varphi(\tau_{i_2}) = 2\Re\varphi(\tau_{i_1})$ in $\mathbb{C}/\Lambda$.

Finally, we map the result z to $E(\mathbb{C})$ via η_E. The point on $E(\mathbb{C})$ will be real and, in theory, rational.

The series (3.32) requires $O(N)$ terms, and converges slowly if the conductor N is large. The algorithm is therefore impractical if $N > 10^8$ [45].

Gross and Zagier's theorem [22] shows that the construction of Heegner points will always provide a point of infinite order on a rank 1 elliptic curve of the height predicted by the Birch and Swinnerton-Dyer conjecture.

Theorem 3.4.1 (Gross and Zagier). *Assume* $L(E, 1) = 0$. *Then there exists a point* P *on* E *such that*

$$L'(E, 1) = \alpha.\Omega.\hat{h}(P) \tag{3.33}$$

with $\alpha \in \mathbb{Q}$.

In particular,

(a) If $L'(E,1) \neq 0$, *then* $E(\mathbb{Q})$ *has rank* ≥ 1.

(b) If $L'(E,1) \neq 0$ *and* $rank(E(\mathbb{Q})) = 1$, *then equation (3.31) is true for some non-zero rational number* α.

Since $L'(E,1)$ can be estimated using standard convergent series, equation (3.33) can be used to compute $\hat{h}(P)$, via the formula

$$\hat{P} = \frac{L'(E,1)}{\alpha\Omega}. \tag{3.34}$$

The point P might be a multiple of a generator, but its height gives an upper bound on the canonical height of a generator.

Further information about Heegner points and their construction is given in [22].

Chapter 4

The Equation $Y^2 = X^3 + pX$, where p is prime

4.1 Overview

Consider the family of elliptic curves

$$Y^2 = X^3 + pX, \tag{4.1}$$

where p is prime, defined over the rational field $\mathbb{Q}$.

Silverman [43] has shown that the rank of the curve (4.1) is given by

$$rank(E) = \begin{cases} 0 & \text{if } p \equiv 7, 11 \pmod{16} \\ 0, 1 & \text{if } p \equiv 3, 5, 13, 15 \pmod{16} \\ 0, 1, 2 & \text{if } p \equiv 1, 9 \pmod{16}. \end{cases}$$

For $p \equiv 5 \pmod{8}$, the Birch and Swinnerton-Dyer conjecture (conjecture 3.3.1) implies that the rank is equal to 1, and Bremner and Cassels [8] have verified this for $p < 1000$ by computing $L'(E, 1)$. Explicit generators were found for all p in the range. Bremner [6] has extended these results to show that the curves have rank 1 for all p in the range $1000 < p < 20000$, and has given explicit generators for $1000 < p < 5000$, with three exceptions (p=3917, 4157, 4957). Co-ordinates for the generators for these exceptions have since been provided by Bremner and Buell [7].

In section 3.2.2, we saw that it is possible to produce a 4-covering $D \in \mathbb{P}^3$ of an elliptic curve E in the form of a pair of simultaneous homogeneous quadratic equations (or *quadric intersection*). The family of curves $E : Y^2 = X^3 + pX$ have a rational 2-torsion point, and so we may use the method of descent via 2-isogeny. The form of the curves makes it possible to perform a fourth descent to find a covering curve D. The map $D \to E$ will be of degree 16, so it is expected that rational points on D will have smaller height than the rational points on E and therefore points on D should be easier to find.

Bremner and Cassel's method, as described in [8], follows.

4.2 Algebraic Preliminaries

Consider the rational points (X, Y) on the curve (4.1). Clearly, $(0, 0) \in E(\mathbb{Q})$. The algebraic structure of the rational points $(X, Y) \neq (0, 0)$ on this curve will now be determined. Since $Y^2 \geq 0$, we must have $X \geq 0$.

By theorem 3.1.4, a rational point $(X, Y) \neq (0, 0)$ on the curve (4.1) is of the form

$$X = \frac{R}{S^2}, \quad Y = \frac{T}{S^3} \tag{4.2}$$

for R, S and $T \in \mathbb{Z}$, with $R \geq 0$ and $\gcd(R, S) = \gcd(S, T) = 1$.

Lemma 4.2.1. *A rational point $(X, Y) \neq (0, 0)$ on the curve (4.1) has X or pX a square.*

Proof. Let $X = \frac{R}{S^2}$, $Y = \frac{T}{S^3}$. Then equation (4.1) becomes

$$\frac{T^2}{S^6} = \frac{R^3}{S^6} + \frac{pR}{S^2},$$

so that

$$T^2 = R^3 + pRS^4. \tag{4.3}$$

Suppose that q is an arbitrary prime such that

$$q^a \parallel R, \quad q^b \parallel S, \quad q^c \parallel T$$

for some integers $a, b, c \geq 0$. Then the exponent of q in $X = \frac{R}{S^2}$ is $(a - 2b)$.

First, consider the case where $q \neq p$.

The left hand side of (4.3) is exactly divisible by q^{2c}. For the right hand side of (4.3), note that R^3 is exactly divisible by q^{3a} and pRS^4 is exactly divisible by q^{a+4b}.

There are then three cases to consider:

1. If $a < 2b$, then $3a < (a + 4b)$ so the right hand side of (4.3) is exactly divisible by q^{3a}. Equating exponents of q in the left and right hand sides of (4.3) gives

   $$2c = 3a \Rightarrow 3a \text{ is even } \Rightarrow a \text{ is even } \Rightarrow (a - 2b) \text{ is even.}$$

2. If $a > 2b$, then $3a > (a + 4b)$ so the right hand side of (4.3) is exactly divisible by q^{a+4b}. Equating exponents of q in the left and right hand sides of (4.3) gives

   $$2c = (a + 4b) \Rightarrow (a + 4b) \text{ is even } \Rightarrow a \text{ is even } \Rightarrow (a - 2b) \text{ is even.}$$

3. If $a = 2b$ then $(a - 2b) = 0$ is even.

So the exponent of q in X is even, for all primes $q \neq p$.

Now suppose that $q = p$. The exponent $(a - 2b)$ of p in $X = \frac{R}{S^2}$ is either even or odd. In the first of these cases, X is a square. In the second case, pX is a square. □

Now consider the addition of points on elliptic curves. For the curve (4.1) if $P_1 \neq -P_2$, formulae (3.11) and (3.12) become

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \quad \nu = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1} \quad \text{if } x_1 \neq x_2$$
$$\lambda = \frac{3x_1^2 + p}{2y_1}, \quad \nu = \frac{-x_1^3 + px_1}{2y_1} \quad \text{if } x_1 = x_2,$$

and then

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = -\lambda x_3 - \nu.$$

Note that if $P_1 = -P_2$, then $P_1 + P_2 = O_E$.

Since $(0, 0) \in E(\mathbb{Q})$, we can consider the pairs of points P and $P + (0, 0)$. Ignoring the point $(0, 0) + (0, 0)$, we have

$$P + (0, 0) = (x_3, y_3) = (x_1, y_1) + (0, 0) = (\frac{p}{x_1}, -\frac{py_1}{x_1^2}).$$

If the point (x_1, y_1) is such that px_1 is a square, then this new point has x_3 a square. So the point (X, Y) can be taken to be (x_3, y_3) and therefore X is a square.

Therefore, it is only necessary to consider points $(X, Y) \neq (0, 0)$ where X is a square.

Lemma 4.2.2. *A rational point $(X, Y) \neq (0, 0)$ with X a square on the curve (4.1) is of the form*

$$X = \frac{r^2}{s^2}, Y = \frac{rt}{s^3} \tag{4.4}$$

for some integers r, s and t with $r \neq 0$, $\gcd(r, s) = \gcd(s, t) = 1$, satisfying

$$t^2 = r^4 + ps^4. \tag{4.5}$$

Proof. By theorem 3.1.4, $X = \frac{R}{s^2}$ and $Y = \frac{T}{s^3}$ for integers R, S and T, with $R \geq 0$ and $\gcd(R, s) = 1$. Since X is a square, R can be written as $R = r^2$ for some integer r. Then $X = \frac{r^2}{s^2}$. Since $\gcd(R, s) = 1$, it follows that $\gcd(r, s) = 1$.

Substituting

$$X = \frac{r^2}{s^2}, \quad Y = \frac{T}{s^3}$$

into the equation for the curve (4.1) gives

$$\begin{aligned} \frac{T^2}{s^6} &= \frac{r^6}{s^6} + p\frac{r^2}{s^2} \\ &= \frac{r^6 + pr^2s^4}{s^6} \\ &= \frac{r^2(r^4 + ps^4)}{s^6}. \end{aligned} \tag{4.6}$$

Equating numerators in (4.6),

$$T^2 = r^2(r^4 + ps^4) \tag{4.7}$$

so that $r^2 \mid T^2$, and therefore $r \mid T$. Let $T = rt$ for some integer t. Then

$$X = \frac{r^2}{s^2}, \quad Y = \frac{rt}{s^3}. \tag{4.8}$$

Equation (4.7) then becomes

$$r^2t^2 = r^2(r^4 + ps^4), \tag{4.9}$$

so that

$$t^2 = r^4 + ps^4. \tag{4.10}$$

Finally, from theorem 3.1.4 it follows that $\gcd(T, s) = 1$, and since $T = rt$ we have $\gcd(rt, s) = 1$. Hence $\gcd(s, t) = 1$. □

Lemma 4.2.3. *In equation (4.4),* $r \not\equiv 0, t \not\equiv 0 \pmod{p}$.

Proof. Suppose that $p \mid r$. Then $t^2 \equiv 0 \pmod{p}$, so that $t \equiv 0 \pmod{p}$. Then $p^2 \mid t^2 = r^4 + ps^4$. But $p \mid r \rightarrow p^2 \mid r^4$. Therefore $p^2 \mid ps^4$ which implies that $p \mid s^4$ and hence $p \mid s$, which contradicts $\gcd(r, s) = 1$. So $p \nmid r$.

If $p \mid t$ then $t^2 \equiv 0 \pmod{p}$, and since $ps^4 \equiv 0 \pmod{p}$ it follows that $r^4 \equiv 0 \pmod{p}$, so that $p \mid r$. But then $p^2 \mid (t^2 - r^4) = ps^4$ which implies that $p \mid s^4$ and hence $p \mid s$, contradicting $\gcd(r, s) = 1$. Hence $p \nmid t$. □

Lemma 4.2.4. *In equation (4.10),* $\gcd(r, s) = \gcd(r, t) = \gcd(s, t) = 1$.

Proof. It has already been shown that $\gcd(r, s) = \gcd(s, t) = 1$. So it remains to be shown that $\gcd(r, t) = 1$.

Suppose $\gcd(r, t) = d \geq 1$. Then we can write

$$r = dr', t = dt',$$

with $\gcd(r', t') = 1$.

Then equation (4.10) becomes

$$d^2t'^2 = d^4r'^4 + ps^4,$$

so that

$$d^2(t'^2 - d^2r'^4) = ps^4. \tag{4.11}$$

Since d^2 divides the left hand side of equation (4.11), it follows that $d^2 \mid ps^4$. But $d \nmid s^4$ as this would contradict $\gcd(r, s) = 1$. Therefore $d^2 \mid p$ and since p is prime, $d = 1$.

Hence $\gcd(r, s) = \gcd(r, t) = \gcd(s, t) = 1$ as required. □

4.3 The Equation $Y^2 = X^3 + pX$, where $p \equiv 5 \pmod 8$

This descent has been performed by Bremner and Cassels [8], who have found rational points via 2-descent for $p < 1000, p \equiv 5 \pmod 8$, and in all cases were able to show that the points found were generators for the group of rational points on the curve. The following is an explanation of their descent method.

It has already been shown that a rational point $(X, Y) \neq (0, 0)$ with X square on the curve (4.1), where p is prime, is of the form

$$X = \frac{r^2}{s^2}, Y = \frac{rt}{s^3}$$

for some pairwise coprime integers r, s and t satisfying

$$t^2 = r^4 + ps^4.$$

We now consider the case where $p \equiv 5 \pmod 8$.

Lemma 4.3.1. *In equation (4.5), we have*

$$r \equiv t \equiv 1 \pmod 2, s \equiv 0 \pmod 2. \tag{4.12}$$

Furthermore, we can choose t such that $t \equiv 1 \pmod 8$.

Proof. A consideration of equation (4.5) modulo 8 leads to a restriction on the parities of r, s and t.

Note that $a^4 \equiv 0, 1 \pmod 8$, according to whether a is even or odd. Since $p \equiv 5 \pmod 8$, $t^2 \equiv r^4 + 5s^4 \pmod 8$. The possible parities of r, s and t are shown in Table 4.1.

r	s	t	$t^2 \pmod 8$	$r^4 + 5s^4 \pmod 8$	conclusion
even	even	even	–	–	contradicts gcd$(r, s) = 1$
even	even	odd	–	–	contradicts gcd$(r, s) = 1$
even	odd	even	0, 4	5	no solutions
even	odd	odd	1	5	no solutions
odd	even	even	0, 4	1	no solutions
odd	even	odd	1	1	✓
odd	odd	even	0, 4	6	no solutions
odd	odd	odd	1	6	no solutions

Table 4.1: Possible parities of r, s and t in equation (4.5)

So the only possible parities of r, s and t are

$$r \equiv t \equiv 1 \pmod 2, s \equiv 0 \pmod 2.$$

Since t is odd, we have $t \equiv \pm 1 \pmod 4$. On taking $-t$ for t where necessary, we can assume that $t \equiv 1 \pmod 4$. From (4.12) we deduce that $r^4 \equiv 1 \pmod{16}$, $s^4 \equiv 0 \pmod{16}$ and $t^2 \equiv 1, 9 \pmod{16}$. But $r^4 + ps^4 \equiv 1 + 0 = 1 \pmod{16}$ so we must have $t^2 \equiv 1 \pmod{16}$. Hence $t \equiv 1, 7, 9, 15 \pmod{16}$ so that $t \equiv \pm 1 \pmod 8$. But $t \equiv 1 \pmod 4$, so that $t \equiv 1, 5 \pmod 8$. Therefore, we must have $t \equiv 1 \pmod 8$. □

Equation (4.5) can be rearranged as

$$ps^4 = t^2 - r^4, \tag{4.13}$$

the right hand side of which can be factorised to give

$$ps^4 = (t - r^2)(t + r^2). \tag{4.14}$$

Since t and r are both odd, each of these two factors must be even. Any prime dividing both $(t - r^2)$ and $(t + r^2)$ would also divide $2t$ and $2r^2$. Suppose d is any such prime, $d \neq 2$, so that $d \mid t$ and $d \mid r^2$, which implies that $d \mid r$. Then $d^2 \mid t^2$ and $d^2 \mid r^4 \implies d^2 \mid (t^2 - r^4) = ps^4$. Now $d^2 \nmid p$ since p is prime, so we must have $d^2 \mid s^4$ so that $d \mid s^2$ and therefore $d \mid s$. This contradicts $\gcd(r, s) = 1$. Hence $\gcd(t - r^2, t + r^2) = 2$.

As s is even, the left hand side of (4.14) is divisible by 16. Since the two factors have greatest common divisor 2, we must have that one factor is divisible by 2 and the other is divisible by 8. Also, exactly one factor is divisible by p. The remaining factor of each must be a fourth power.

Since $t \equiv 1 \pmod 8$ and $r^2 \equiv 1 \pmod 8$, we must have $t + r^2 \equiv 2 \pmod 8$ and $t - r^2 \equiv 0 \pmod 8$. The four possibilities are given in Table 4.2.

Case	$t + r^2$	$t - r^2$
(A)	$2pa^4$	$8b^4$
(B)	$2a^4$	$8pb^4$
(C)	$-2pa^4$	$-8b^4$
(D)	$-2a^4$	$-8pb^4$

Table 4.2: The four possibilities for equation (4.14)

Case (B) gives $t = a^4 + 4pb^4 \equiv 1 + 20 = 21 \equiv 5 \pmod 8$.

Case (C) gives $t = -pa^4 - 4b^4 \equiv -5 - 4 = -9 \equiv 7 \pmod 8$.

Case (D) gives $t = -a^4 - 4pb^4 \equiv -1 - 20 = -21 \equiv 3 \pmod 8$.

So the only possibility is case (A). Then we have

$$t + r^2 = 2pa^4, t - r^2 = 8b^4.$$

This gives

$$t = pa^4 + 4b^4, r^2 = pa^4 - 4b^4, s = 2ab. \tag{4.15}$$

Since $p \equiv 5 \pmod 8$, we can write $p = u^2 + 4v^2$ with u,v odd and $\gcd(u, v) = 1$.

We can choose the sign of v so that $v \equiv 1 \pmod 4$. The values of u and v are found using the algorithm outlined in subsection 4.3.1.

From equation (4.15), we have that $r^2 = pa^4 - 4b^4$, or

$$pa^4 = (u^2 + 4v^2)a^4 = r^2 + 4b^4.$$

Factoring over $\mathbb{Z}[i]$ gives

$$(r + 2ib^2)(r - 2ib^2) = (u + 2iv)(u - 2iv)a^4.$$

Note that $(u+2iv)$ is prime. We can choose the sign of r so that $(u+2iv) \mid (r+2ib^2)$

(adjusting the value of s if necessary). Then

$$r + 2ib^2 = (u + 2iv).unit.(c + id)^4 \tag{4.16}$$

with $c^2 + d^2 = a$, and without loss of generality,

$$c \equiv 0 \pmod 2, d \equiv 1 \pmod 2.$$

The units in $\mathbb{Z}[i]$ are $\pm 1, \pm i$, so the possibilities are

(A) $r + 2ib^2 = (u + 2iv)(c + id)^4$

(B) $r + 2ib^2 = -(u + 2iv)(c + id)^4$

(C) $r + 2ib^2 = i(u + 2iv)(c + id)^4$

(D) $r + 2ib^2 = -i(u + 2iv)(c + id)^4$

We consider the imaginary parts of these equations modulo 8. We have $b^2 \equiv 1 \pmod 8$ since b is odd, $u \equiv 1, 3, 5, 7 \pmod 8$ since u is odd, and $v \equiv 1, 5 \pmod 8$ since $v \equiv 1 \pmod 4$. Then $2v \equiv 2 \pmod 8$. Also, c and d are of opposite parity, so that $c^2d^2 \equiv 0, 4 \pmod 8$ and $c^4 + d^4 \equiv 1 \pmod 8$.

The imaginary part of the left hand side in each possibility is $2b^2 \equiv 2 \pmod 8$.

(A) $\text{Im(RHS)} \equiv 2v(c^4 + d^4 - 6c^2d^2) \equiv 2 \pmod 8$

(B) $\text{Im(RHS)} \equiv -2v(c^4 + d^4 + 6c^2d^2) \equiv 6 \pmod 8$

(C) $\text{Im(RHS)} \equiv u(c^4 + d^4 - 6c^2d^2) \equiv 1, 3, 5, 7 \pmod 8$

(D) $\text{Im(RHS)} = \equiv -u(c^4 + d^4 - 6c^2d^2) \equiv 1, 3, 5, 7 \pmod 8$

So the only possibility is that the unit in (4.16) is 1.

Then equating imaginary parts, we have

$$\begin{aligned} 2b^2 &= 4uc^3d - 4ucd^3 + 2vc^4 - 12vc^2d^2 + 2vd^4 \\ &= 2v(c^4 - 6c^2d^2 + d^4) + 2u(2c^3d - 2cd^3), \end{aligned}$$

so that

$$b^2 = v(l^2 - m^2) + ulm, \tag{4.17}$$

where

$$l = c^2 - d^2, m = 2cd. \tag{4.18}$$

If there are solutions to (4.1), where $p \equiv 5 \pmod 8$, then there must be rational solutions to (4.17). These rational solutions exist for all primes $p \equiv 5 \pmod 8$ such that $p < 5000$ [8], and every such solution is given by one of a finite number of parameterisations

$$l = q_1(\theta, \psi), \quad m = q_2(\theta, \psi), \quad b = q_3(\theta, \psi), \tag{4.19}$$

where q_1, q_2 and q_3 are quadratic forms with integer coefficients and θ and ψ are integers to be found.

Then

$$(c + id)^2 = l + im = q_1(\theta, \psi) + iq_2(\theta, \psi). \tag{4.20}$$

Again, all solutions to (4.20) are given by a unique parameterisation in $\mathbb{Z}[i]$

$$\theta = Q_1(\lambda, \mu), \quad \psi = Q_2(\lambda, \mu), \quad (c + id) = Q_3(\lambda, \mu), \tag{4.21}$$

where Q_1, Q_2 and Q_3 are quadratic forms with coefficients in $\mathbb{Z}[i]$ and $\lambda, \mu \in \mathbb{Z}[i]$.

The parameterisations (4.19) and (4.21) are found using the procedure outlined in subsection 4.3.2.

Now θ and ψ are (rational) integers, so $Im(\theta) = Im(\psi) = 0$. If we have $\lambda = \alpha + i\beta$ and $\mu = \gamma + i\delta$, then this condition leads to a pair of simultaneous quadratic equations in the four variables $\alpha, \beta, \gamma, \delta$.

A fully worked example of this descent process will be given in chapter 7. Methods for solving the simultaneous equations will be discussed in chapter 6.

4.3.1 Finding u and v such that $p = u^2 + 4v^2$

The existence of such a u and v depends on Dirichlet's Approximation Theorem.

Theorem 4.3.1 (Dirichlet's Approximation Theorem). *Let θ be any real number, and let L be any natural number. Then there exist integers a and b such that* $1 \leq b \leq L$ *and*

$$\left|\theta - \frac{a}{b}\right| \leq \frac{1}{(L+1)b}$$

As $p \equiv 5 \pmod 8$, by theorem 2.4.1(c) and (d) we have

$$\left(\frac{-2}{p}\right) = -1,$$

and by Euler's Criterion (theorem 2.4.2),

$$-1 \equiv (-2)^{(p-1)/2} \pmod p.$$

But $p = 8n + 5$ for some integer n, so $(p-1)/2 = 4n + 2$. Then

$$-1 \equiv (-2)^{4n+2} = \left((-2)^{2n+1}\right)^2 \pmod p.$$

Let $j \equiv (-2)^{2n+1} \pmod p$, so that $j^2 \equiv -1 \pmod p$. Taking $\theta = \frac{j}{p}$ and $L = \lfloor \sqrt{p} \rfloor$ in Dirichlet's Approximation Theorem (theorem 4.3.1), there exist integers v and

w such that $1 \leq w \leq \lfloor \sqrt{p} \rfloor$ and

$$\left| \frac{j}{p} - \frac{v}{w} \right| \leq \frac{1}{\left(\lfloor \sqrt{p} \rfloor + 1 \right) w}.$$

Since p is prime, we have $1 \leq w < \sqrt{p}$ and

$$\left| \frac{j}{p} - \frac{v}{w} \right| < \frac{1}{w\sqrt{p}}.$$

Let $u = jw - pv$. Then

$$\left| \frac{j}{p} - \frac{v}{w} \right| = \left| \frac{u}{pw} \right| < \frac{1}{w\sqrt{p}} \implies |u| < \sqrt{p}.$$

Now

$$\begin{aligned} u^2 + w^2 &= (jw - pv)^2 + w^2 \\ &\equiv (jw)^2 + w^2 \pmod{p} \\ &\equiv (j^2 + 1)w^2 \pmod{p}. \end{aligned}$$

But $j^2 \equiv -1 \pmod{p}$, so $u^2 + w^2 \equiv 0 \pmod{p}$. Thus $u^2 + w^2$ is an integer multiple of p. Since $1 \leq w < \sqrt{p}$ and $|u| < \sqrt{p}$, we have $0 < u^2 + w^2 < 2p$. So the only possibility is that $u^2 + w^2 = p$.

Now u and w cannot be of the same parity, otherwise p would be even. So we can assume that u is odd and w is even. Let $w = 2v$. Then $p = u^2 + 4v^2$, with u odd and $\gcd(u, v) = 1$. If v is even, then $v = 2z$ gives $p = u^2 + 16z^2 \equiv u^2 \pmod{8}$. But $p \equiv 5 \pmod{8}$ and $u^2 \equiv 5 \pmod{8}$ has no solutions. Therefore v must also be

odd.

Hence $p = u^2 + 4v^2$, with u, v odd and $\gcd(u, v) = 1$.

The set $\{(u, w) : u \equiv jw \pmod{p}\}$ forms a lattice spanned by the lattice basis $\langle (j, 1), (p, 0) \rangle$. By using a suitable lattice reduction algorithm (for example the LLL algorithm), we can take (u, w) to be the shortest vector in this lattice. Due to the properties of reduced lattice bases (see section 5.2 for details), the second vector in the reduced basis is guaranteed to be short enough. Note that we may need to swap the elements of this vector to endure that u is odd.

4.3.2 Finding an equivalent quadratic form

Suppose we are considering the equation

$$Q(x_1, x_2, x_3) = 0, \tag{4.22}$$

where

$$Q(x_1, x_2, x_3) = K_1x_1^2 + K_2x_2^2 + K_3x_3^2 + K_4x_1x_2 + K_5x_1x_3 + K_6x_2x_3$$

is a quadratic form in the three variables x_1, x_2 and x_3. Suppose that a solution (a_1, a_2, a_3) to equation (4.22) is known, with $a_i \in \mathbb{Z}$ and $\gcd(a_1, a_2, a_3) = 1$.

We first find a matrix

$$M = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

with $b_i, c_i \in \mathbb{Z}$, such that $\det(M) = 1$.

The substitution

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

is then made. Since $\det(M) = 1$, the transformation is unimodular.

Then

$$\begin{aligned}
Q(x_1, x_2, x_3) = {} & X_1^2[K_1a_1^2 + K_2a_2^2 + K_3a_3^2 + K_4a_1a_2 + K_5a_1a_3 + K_6a_2a_3] \\
& + X_2^2[K_1b_1^2 + K_2b_2^2 + K_3b_3^2 + K_4b_1b_2 + K_5b_1b_3 + K_6b_2b_3] \\
& + X_3^2[K_1c_1^2 + K_2c_2^2 + K_3c_3^2 + K_4c_1c_2 + K_5c_1c_3 + K_6c_2c_3] \\
& + X_1X_2[2K_1a_1b_1 + 2K_2a_2b_2 + 2K_3a_3b_3 + K_4(a_1b_2 + a_2b_1) \\
& \qquad + K_5(a_1b_3 + a_3b_1) + K_6(a_2b_3 + a_3b_2)] \qquad (4.23) \\
& + X_1X_3[2K_1a_1c_1 + 2K_2a_2c_2 + 2K_3a_3c_3 + K_4(a_1c_2 + a_2c_1) \\
& \qquad + K_5(a_1c_3 + a_3c_1) + K_6(a_2c_3 + a_3c_2)] \\
& + X_2X_3[2K_1b_1c_1 + 2K_2b_2c_2 + 2K_3b_3c_3 + K_4(b_1c_2 + b_2c_1) \\
& \qquad + K_5(b_1c_3 + b_3c_1) + K_6(b_2c_3 + b_3c_2)].
\end{aligned}$$

The coefficient of X_1^2 in (4.23) is $Q(a_1, a_2, a_3) = 0$.

So let

$$\begin{aligned}
H(X_1, X_2, X_3) &= Q(x_1, x_2, x_3) \\
&= dr_2X_1X_2 + dr_3X_1X_3 + 4r_4X_2^2 + r_5X_2X_3 + r_6X_3^2,
\end{aligned} \qquad (4.24)$$

where

$$r_2 = [2K_1a_1b_1 + 2K_2a_2b_2 + 2K_3a_3b_3 + K_4(a_1b_2 + a_2b_1)$$
$$+ K_5(a_1b_3 + a_3b_1) + K_6(a_2b_3 + a_3b_2)]/d$$

$$r_3 = [2K_1a_1c_1 + 2K_2a_2c_2 + 2K_3a_3c_3 + K_4(a_1c_2 + a_2c_1)$$
$$+ K_5(a_1c_3 + a_3c_1) + K_6(a_2c_3 + a_3c_2)]/d$$

$$r_4 = K_1b_1^2 + K_2b_2^2 + K_3b_3^2 + K_4b_1b_2 + K_5b_1b_3 + K_6b_2b_3$$

$$r_5 = 2K_1b_1c_1 + 2K_2b_2c_2 + 2K_3b_3c_3 + K_4(b_1c_2 + b_2c_1)$$
$$+ K_5(b_1c_3 + b_3c_1) + K_6(b_2c_3 + b_3c_2)$$

$$r_6 = K_1c_1^2 + K_2c_2^2 + K_3c_3^2 + K_4c_1c_2 + K_5c_1c_3 + K_6c_2c_3,$$

with $\gcd(r_2, r_3) = 1$.

Then

$$H(X_1, X_2, X_3) = X_1(dr_2X_2 + dr_3X_3) + r_4X_2^2 + r_5X_2X_3 + r_6X_3^2$$
$$= X_1L(X_2, X_3) + H_0(X_2, X_3),$$

where L is a linear form and H_0 is a quadratic form, each in the two variables X_2 and X_3.

We now find $s_2, s_3 \in \mathbb{Z}$ such that $r_2s_3 - r_3s_2 = 1$, and make the substitution

$$\begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_2 & r_3 \\ 0 & s_2 & s_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}.$$

The transformation is unimodular since the matrix has determinant 1.

Then

$$H(X_1, X_2, X_3) = dY_1Y_2 + H_1(Y_2, Y_3),$$

where $H_1(Y_2, Y_3) = aY_2^2 + bY_2Y_3 + cY_3^2$ is a quadratic form in the variables Y_2 and Y_3.

So now we have

$$\begin{aligned} H &= dX_1(r_2X_2 + r_3X_3) + a(r_2X_2 + r_3X_3)^2 \\ &\quad + b(r_2X_2 + r_3X_3)(s_2X_2 + s_3X_3) + c(s_2X_2 + s_3X_3)^2 \\ &= dr_2X_1X_2 + dr_3X_1X_3 + X_2^2(ar_2^2 + br_2s_2 + cs_2^2) \\ &\quad + X_3^2(ar_3^2 + br_3s_3 + cs_3^2) + X_2X_3(2ar_2r_3 + br_2s_3 + br_3s_2 + 2cs_2s_3). \end{aligned} \tag{4.25}$$

Equating coefficients of X_2^2, X_2X_3, X_3^2 in equations (4.25) and (4.24) gives

$$\begin{aligned} ar_2^2 + br_2s_2 + cs_2^2 &= r_4 \\ ar_3^2 + br_3s_3 + cs_3^2 &= r_6 \\ 2ar_2r_3 + br_2s_3 + br_3s_2 + 2cs_2s_3 &= r_5, \end{aligned}$$

and these equations can be solved for a, b and c.

Finally, let

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} d & a & b \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix}.$$

This transformation will be unimodular if and only if $d = 1$.

Then

$$Z_1Z_2 + cZ_3^2 = (dY_1 + aY_2 + bY_3)Y_2 + cY_3^2$$

$$= dY_1Y_2 + aY_2^2 + bY_2Y_3 + cY_3^2.$$

So the equation

$$Z_1Z_2 + cZ_3^2 = 0$$

is equivalent to equation (4.22) as required.

4.4 Maple Worksheet

A Maple worksheet to perform this descent process is included in Appendix A. The worksheet creates two text files. The first contains the two quadratic equations to be solved, together with the search range to be used. The second file contains all the intermediate equations, in order to be able to find the generator of the points on the curve given the solution to the pair of simultaneous equations.

Chapter 5

Lattices and Lattice Reduction

5.1 Overview

Definition 5.1.1. A *(point) lattice* $\mathcal{L}$ of *dimension* n is the set of all integral linear combinations of a set of n basis vectors in $\mathbb{R}^m$.

Therefore we can write

$$\mathcal{L} = \left\{ \boldsymbol{x} = (x_1, x_2, \ldots, x_m) : x = \sum_{i=1}^{n} \lambda_i \boldsymbol{b}_i, \lambda_i \in \mathbb{Z} \right\},$$

where $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ are linearly independent vectors in $\mathbb{R}^m$.

Definition 5.1.2. The spanning set of vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n$ forms a *basis* for the lattice $\mathcal{L}$.

This basis can be written in matrix form as

$$B = [\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_n}],$$

so that B is an $m \times n$ matrix.

A lattice may have several bases. If C is also a basis for the lattice $\mathcal{L}$, then there exists an $m \times m$ matrix U with determinant ± 1 such that

$$C = UB.$$

Definition 5.1.3. The *determinant* $\Delta(\mathcal{L})$ of the lattice $\mathcal{L}$ is given by

$$\Delta(\mathcal{L}) = \left|\det\left(B^T B\right)\right|^{\frac{1}{2}}.$$

From definition 5.1.3, it can be seen that the determinant of the lattice does not depend on the basis, since

$$\begin{aligned}
\Delta(\mathcal{L}) &= \left|\det\left(C^T C\right)\right|^{\frac{1}{2}} \\
&= \left|\det\left((UB)^T (UB)\right)\right|^{\frac{1}{2}} \\
&= \left|\det\left(B^T U^T UB\right)\right|^{\frac{1}{2}} \\
&= \left|\det\left(B^T\right)\det\left(U^T\right)\det(U)\det(B)\right|^{\frac{1}{2}} \\
&= \left|\det\left(B^T\right)\det(B)\right|^{\frac{1}{2}} \\
&= \left|\det\left(B^T B\right)\right|^{\frac{1}{2}}.
\end{aligned}$$

Also note that if $n = m$, then $\Delta(\mathcal{L}) = |\det B|$.

Although all bases for a lattice have the same determinant, one basis might be considered better than another if its vectors are orthogonal with respect to the inner product, or if it contains shorter vectors than the other. An orthogonal basis for a vector space can be found using the Gram-Schmidt process.

Theorem 5.1.4 (Gram-Schmidt). *A vector space with a basis* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ *and inner product* $\langle , \rangle$ *has an orthogonal basis given by* $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ *where*

$$\boldsymbol{b}_i^* = \boldsymbol{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \boldsymbol{b}_j^*, \quad i = 1, \ldots, n,$$

and the $\mu_{i,j}$ *are defined by*

$$\mu_{i,j} = \frac{\langle \boldsymbol{b}_i, \boldsymbol{b}_j^* \rangle}{\langle \boldsymbol{b}_j^*, \boldsymbol{b}_j^* \rangle}.$$

However, since the $\mu_{i,j}$ may not be integers, the new basis $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ might not be a basis for the lattice spanned by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.

It might be that no orthogonal basis *exists* for a particular lattice, and so it is desirable to find a lattice basis which is almost orthogonal.

5.2 LLL-reduced Lattice Bases

Suppose that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is a basis for the lattice $\mathcal{L}$, and that $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ is the basis produced by the Gram-Schmidt process (Theorem 5.1.4).

Definition 5.2.1. The basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is called *LLL-reduced* if

$$|\mu_{i,j}| \leq \frac{1}{2}, \qquad 1 \leq j < i \leq n$$

and

$$\left\|\boldsymbol{b}_i^* + \mu_{i,i-1}\boldsymbol{b}_{i-1}^*\right\|^2 \geq \frac{3}{4}\left\|\boldsymbol{b}_{i-1}^*\right\|^2, \qquad 1 < i \leq n.$$

The first part of the definition ensures that the vectors $\boldsymbol{b}_i$ are almost orthogonal, while the second part restricts the relative sizes of the basis vectors. This is equivalent to

$$\left\|\boldsymbol{b}_i^*\right\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right)\left\|\boldsymbol{b}_{i-1}^*\right\|^2$$

since the $\boldsymbol{b}_i^*$ are orthogonal.

LLL-reduced bases have several desirable properties, including the following.

Theorem 5.2.2. *If $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is an LLL-reduced basis for the lattice $\mathcal{L}$, then*

(a) $\left\|\boldsymbol{b}_j\right\|^2 \leq 2^{i-1}\left\|\boldsymbol{b}_i^*\right\|^2$, *for* $1 \leq j \leq i \leq n$

(b) $\Delta(\mathcal{L}) \leq \prod_{i=1}^{n} \|\boldsymbol{b}_i\| \leq 2^{n(n-1)/4}\Delta(\mathcal{L})$

Proof. (a) $\left\|\boldsymbol{b}_i^*\right\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right)\left\|\boldsymbol{b}_{i-1}^*\right\|^2 \geq \frac{1}{2}\left\|\boldsymbol{b}_{i-1}^*\right\|^2$ for $i = 2, \ldots, n$. Then

$\left\|\boldsymbol{b}_j^*\right\|^2 \leq 2^{i-j}\left\|\boldsymbol{b}_i^*\right\|^2$ for $1 \leq j \leq i \leq n$, by induction, so that

$$\begin{aligned}\|\boldsymbol{b}_i\|^2 &= \left\|\boldsymbol{b}_i^*\right\|^2 + \sum_{j=1}^{i-1} \mu_{i,j}^2 \left\|\boldsymbol{b}_j^*\right\|^2 \\ &\leq \left(1 + \sum_{j=1}^{i-1} 2^{i-j-2}\right)\left\|\boldsymbol{b}_i^*\right\|^2 \\ &= \left(1 + \frac{1}{4}(2^i - 2)\right)\left\|\boldsymbol{b}_i^*\right\|^2 \\ &\leq 2^{i-1}\left\|\boldsymbol{b}_i^*\right\|^2 .\end{aligned} \tag{5.1}$$

Hence $\left\|\boldsymbol{b}_j\right\|^2 \leq 2^{j-1}\left\|\boldsymbol{b}_j^*\right\|^2 \leq 2^{j-1+i-j}\left\|\boldsymbol{b}_i^*\right\|^2 = 2^{i-1}\left\|\boldsymbol{b}_i^*\right\|^2$.

(b) Writing $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ in terms of $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$, we have

$$B = B^* J,$$

where

$$B = [\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n],$$

$$B^* = \left[\boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \ldots, \boldsymbol{b}_n^*\right],$$

$$J = \begin{pmatrix} 1 & \mu_{2,1} & \mu_{3,1} & \cdots & \mu_{n,1} \\ 0 & 1 & \mu_{3,2} & \cdots & \mu_{n,2} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \mu_{n,n-1} \\ 0 & \cdots & \cdots & \cdots & 1 \end{pmatrix}.$$

Then

$$\begin{aligned}\Delta(\mathcal{L})^2 &= \det(B^T B) \\ &= \det\left(J^T (B^*)^T B^* J\right) \\ &= \det\left((B^*)^T B^*\right) \\ &= \prod_{i=1}^{n} (\boldsymbol{b}_i^*)^T \boldsymbol{b}_i^* \\ &= \prod_{i=1}^{n} \left\|\boldsymbol{b}_i^*\right\|^2 .\end{aligned}$$

From the first equality of (5.1), we have $\left\|\boldsymbol{b}_i^*\right\| \leq \|\boldsymbol{b}_i\|$, so that

$$\begin{aligned}\Delta(\mathcal{L}) &\leq \prod_{i=1}^{n} \|\boldsymbol{b}_i\| \\ &\leq \prod_{i=1}^{n} 2^{(i-1)/2} \left\|\boldsymbol{b}_i^*\right\| \\ &= 2^{n(n-1)/4} \prod_{i=1}^{n} \left\|\boldsymbol{b}_i^*\right\| \\ &= 2^{n(n-1)/4} \Delta(\mathcal{L}).\end{aligned}$$

□

5.3 The LLL Algorithm

The following algorithm was devised by Lenstra, Lenstra and Lovász in 1982 [25].

Given a basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ for the lattice $\mathcal{L}$, we wish to find a basis which satisfies definition 5.2.1.

We first find the associated Gram-Schmidt basis $\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_n^*$ and the constants $\mu_{i,j}$ using theorem 5.1.4.

Let $k = 2$. If $n = 1$ do nothing, else perform the following steps.

Step 1

Reduce $|\mu_{k,k-1}|$ to at most $\frac{1}{2}$ by adding a suitable multiple of $\boldsymbol{b}_{k-1}$ to $\boldsymbol{b}_k$.

Step 2

If the second part of definition 5.2.1 holds with $i = k$, then go to Step 3. Otherwise, swap vectors $\boldsymbol{b}_{k-1}$ and $\boldsymbol{b}_k$, making the necessary changes to the Gram-Schmidt basis and the $\mu_{i,j}$. If $k > 2$, subtract 1 from k. Go to step 1.

Step 3

For $j = k-2, k-3, \ldots, 1$, reduce $|\mu_{k,j}|$ to at most $\frac{1}{2}$. Add 1 to k. If $k = n+1$ the algorithm terminates, otherwise go to step 1.

The algorithm terminates when the value of k reaches $n+1$. However, as well as being increased in step 3, the value of k is reduced in step 2. To show that the algorithm must eventually terminate requires the following theorem.

Theorem 5.3.1 (Hermite). *There exist positive real constants μ_n (for $n = 1, 2, \ldots$), depending only on n, such that for any lattice $\mathcal{L}$ of dimension n,*

$$M_1^n \leq \mu_n \Delta(\mathcal{L})^2,$$

where M_1 is the square of the minimum length of a non-zero vector in the lattice $\mathcal{L}$.

An important corollary of Hermite's theorem is the following.

Corollary 5.3.2. *If $\mathcal{L}$ is a lattice of dimension n, then there exists a constant $c > 0$, depending only on $\mathcal{L}$, such that for any sublattice $\mathcal{L}_1$ of $\mathcal{L}$, $\Delta(\mathcal{L}_1) \geq c$.*

Now, define $\mathcal{L}_i$ to be the sublattice spanned by the vectors $\boldsymbol{b_1}, \ldots, \boldsymbol{b_i}$ where $1 \leq i \leq n$. Steps 1 and 3 do not change any of the $\mathcal{L}_i$. So we need only consider step 2.

If a swap is performed, the sublattice $\mathcal{L}_{k-1}$ will be changed, while the other sub-

lattices remain unchanged. The new vector $\boldsymbol{b}^*_{k-1}$ is given by

$$\boldsymbol{b}^*_{k-1} \longleftarrow \boldsymbol{b}^*_k + \mu_{k,k-1}\boldsymbol{b}^*_{k-1}.$$

The swap is performed only if the second part of definition 5.2.1 is not satisfied, and the swap multiples $\left\|\boldsymbol{b}^*_{k-1}\right\|^2$ by a factor of less than $\frac{3}{4}$.

But the determinant of the sublattice $\mathcal{L}_{k-1}$ is given by

$$\begin{aligned}\Delta(\mathcal{L}_{k-1})^2 &= \det\left([\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{k-1}]^T\,[\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{k-1}]\right)\\ &= \det\left(\left[\boldsymbol{b}^*_1,\ldots,\boldsymbol{b}^*_{k-1}\right]^T\left[\boldsymbol{b}^*_1,\ldots,\boldsymbol{b}^*_{k-1}\right]\right)\\ &= \prod_{i=1}^{k-1}\left\|\boldsymbol{b}^*_i\right\|^2.\end{aligned}$$

So we see that performing a swap in step 2 will have the effect of multiplying $\Delta(\mathcal{L}_{k-1})^2$ by a factor of less then $\frac{3}{4}$, while the other $\Delta(\mathcal{L}_i)$ remain unchanged.

By corollary 5.3.2, the $\Delta(\mathcal{L}_i)$ are bounded below by a constant depending only on $\mathcal{L}$. The swap can therefore be performed only finitely many times, so the algorithm must terminate.

Note that the constant $\frac{3}{4}$ in step 2 can be replaced by any constant c such that $\frac{1}{4} < c \leq 1$. This would affect the second part of definition 5.2.1, and also the constants in theorem 5.2.2. Rather than 2, the constants in the theorem would become $\alpha = \frac{1}{(c-\frac{1}{4})}$. Taking $c = 1$ would guarantee that the first vector in the basis is the minimal vector in the lattice. However, the algorithm is faster with a smaller

value of c.

Lenstra, Lenstra and Lovász showed that for a lattice containing n basis vectors in $\mathbb{R}^n$, in which the maximum length of a basis vector is M, an upper bound for the number of iterations needed for all values of c except 1 is given by $n^2 \log_{\delta} M + n$, where $c = \frac{1}{\delta^2}$. However, Akhavi has since shown that for a lattice of fixed dimension, the number of iterations for the optimal algorithm (when $c = 1$) is linear with respect to the size of the input vectors [1].

Since we will use the LLL algorithm for lattices in $\mathbb{Z}^n$, the $\mu_{i,j}$ and the vector components throughout the algorithm will be rational. However, the numerators and denominators may become too large to be practical. de Weger's variant of the LLL algorithm uses only integer arithmetic [51]. The algorithm runs in polynomial time and the coefficients do not grow prohibitively large. Pseudocode for the algorithm is given in [46] and [10].

Chapter 6

The Solution of Pairs of Simultaneous Quadratic Equations

6.1 Overview

The method of descent outlined in chapter 4 produces a pair of simultaneous homogeneous quadratic equations in four variables. Methods for solving these simultaneous equations are presented in this chapter. For a given search bound N, the most naïve search would take time $O(N^4)$. A new method, with running time $O(N^{2/3})$ will be presented in section 6.3. Variations on the method, including a method for searching for solutions to a homogeneous ternary quartic equation with running time $O(N)$, will also be given.

Suppose we wish to search for solutions of the simultaneous quadratic equations

$$Q_1(\boldsymbol{w}) = Q_2(\boldsymbol{w}) = 0, \tag{6.1}$$

with $\boldsymbol{w} \in \mathbb{Z}^4$ in the region $| w_i | \leq N$. Since the equations are homogeneous, we may restrict ourselves to looking for coprime solutions.

An upper bound on the height of a generator can be found using equation (3.34),

$$\hat{h}(P) = \frac{L'(E,1)}{\alpha\Omega}.$$

For the equation $Y^2 = X^3 + pX$, Bremner [6] has shown that $\alpha = 2$ and

$$\begin{aligned}
\Omega &= \int_{E(\mathbb{R})} |\omega| \\
&= \frac{1}{4}\int_0^\infty \frac{dx}{\sqrt{x^3 + px}} \\
&= \frac{1}{2p^{1/4}}\int_0^\infty \frac{dt}{\sqrt{t^4 + 1}} \\
&\approx \frac{1}{p^{1/4}}(0.92703733848).
\end{aligned}$$

Therefore,

$$\hat{h}(P) \approx 0.5393526013 p^{1/4} L'(E,1).$$

Since the descent process outlined in chapter 4 is a degree 16 map, an approximate bound on the size of the coordinates of the solutions to the simultaneous equations can be calculated as $N = e^{\hat{h}(P)/16}$.

Having calculated this bound, we can simply try each possible combination of integers within this range. If the range is $(2N + 1)$ (i.e.: each variable lies between $-N$ and N) then this method will need $(2N + 1)^4$ trials.

A second method involves eliminating one of the variables and searching on the other three. Suppose $Q_1(x, y, w, z)$ has no x^2 term. This can be achieved by multiplying each of the two simultaneous equations by the coefficient of x^2 in the other. One equation is then subtracted from the other to leave an equation with no x^2 term.

Then we can write

$$Q_1 = x.L(y, w, z) + Q(y, w, z) = 0,$$

where $L(y, w, z)$ is a linear expression and $Q(y, w, z)$ is a quadratic expression.

Then

$$x.L(y, w, z) = -Q(y, w, z)$$

$$\Longrightarrow \quad x = -\frac{Q(y, w, z)}{L(y, w, z)}$$

and

$$Q_2(x, y, w, x) = Q_2\left(-\frac{Q(y, w, z)}{L(y, w, z)}, y, w, z\right)$$

$$= g(y, w, z).$$

We can then use an exhaustive search on the three variables y, w and z. This requires $(2N + 1)^3$ trials.

The standard, resultant-based method is to let (x, y) run through pairs of coprime integers, substituting the values into Q_1 and Q_2. We then calculate the resultants

Res_w and Res_z and solve the polynomial equations to find possible integer roots for w and z. This will require $(2N + 1)^2$ trials.

These three naïve searches have been implemented in Maple, and results are presented in chapter 7.

More efficient searches can be developed using a version of the method due to Elkies described below.

6.2 The Elkies method

Elkies [18] gives an algorithm for finding rational points of small height *near* a plane curve C. The algorithm uses linear approximations and lattice reduction, and can be adapted to finding points *on* C by embedding the curve into a projective space of higher dimension. The algorithm, and the adaptation for finding points on the curve, are described below.

We assume that the algebraic plane curve C is the image of a differentiable map $\phi : [0, 1] \to \mathbb{RP}^2$ and fix a positive $\delta \leq 1$ such that $\delta \gg N^{-2}$. We will find all rational points of height at most N in $\mathbb{P}^2$ which are at a distance at most δ from C.

The interval $[0, 1]$ is partitioned into $O(\delta^{-1/2})$ intervals I_m, each of length $|I_m| = O(\delta^{1/2})$. On each interval I_m, we use a linear approximation $\overline{\phi}$ to approximate the map ϕ to within $O(|I_m|) = O(\delta)$. Then any point at a distance at most δ from $\phi(I_m)$ will be at a distance $\ll \delta$ from $\overline{\phi}(I_m)$.

Around each $\phi(I_m)$, we construct a parallelepiped with height, length and width proportional to N, $\delta^{1/2}N$ and δN. The points $(x : y : z) \in \mathbb{P}^2$ of height at most N and within a distance $O(\delta)$ of $\phi(I_m)$ will be non-zero integer points in this parallelepiped. So we wish to find all points in $P_m \cap \mathbb{Z}^3$. Since $\delta \gg N^{-2}$, the volume of the parallelepiped is $\gg 1$. Therefore, we expect $|P_m \cap \mathbb{Z}^3|$ to be approximately the volume of P_m. We use lattice reduction techniques to list the points.

To do this, we let M_n be an invertible 3×3 matrix, where $M_m P_m$ is the cube $K = [-1, 1]^3$. To find the points in $P_m \cap \mathbb{Z}^3$, we need to find all of the vectors in

$K \cap M_m^{-1}\mathbb{Z}^3$, so we reduce the lattice $M_m^{-1}\mathbb{Z}^3$ to obtain a matrix $L_m \in \mathrm{GL}_3(\mathbb{Z})$. The vector $\boldsymbol{v}$ is in $K \cap M_m^{-1}\mathbb{Z}^3$ if and only if $\boldsymbol{w} \in \mathbb{Z}^3 \cap (M_m L_m)^{-1}K$ where $\boldsymbol{v} = L_m \boldsymbol{w}$. To find these vectors $\boldsymbol{w}$, we note that $(M_m L_m)^{-1}K$ is contained in the box whose i^{th} side is twice the l^1 norm of the i^{th} row of $(M_m L_m)^{-1}$, where $i = 1, 2, 3$. For each non-zero integral $\boldsymbol{w}$ in this box, we calculate $(x, y, z) = L_m \boldsymbol{w}$ and test whether the point $(x : y : z)$ is of the correct height and distance from C.

To find rational points which lie on the curve C, we first embed C in $\mathbb{P}^M$ where $M > 2$. A segment of the curve in $\mathbb{P}^M$ of length $\ll N^{-2/M}$ will then be contained in a box whose i^{th} side is $\ll N^{-2i/M}$, where $i = 1, 2, \ldots, M$. The rational points of height at most N in this box are the points of $\mathbb{Z}^{M+1}$ contained in the box B whose i^{th} side is $\ll N^{1-2i/M}$ for $i = 0, 1, 2, \ldots, M$. The volume of the box B is $O(1)$. To find the points, we use lattice reduction as before. The result of the reduction will either give a list of points in $\mathbb{Z}^{M+1} \cap B$, or a hyperplane containing $\mathbb{Z}^{M+1} \cap B$. If the latter is the case, we map the hyperplane to $\mathbb{P}^M$ and intersect it with C. The points thus found are then tested to determine whether they lie on the curve.

Higher descents on an elliptic curve E embed the curve as an intersection of two quadrics in $\mathbb{P}^3$, so this method could be used to find points in $E(\mathbb{Q})$. Due to the difficulty in optimising the number and volume of the parallelepipeds to be used, Womack [53] devised a version of the method using p-adic approximations to E rather than the real approximations suggested by Elkies. Womack's method is as follows.

For a search bound N, we first choose a prime P slightly larger than N. For each x_2 such that $1 \leq x_2 \leq P$, we list the solutions $\boldsymbol{x} \in \mathbb{Z}_P^4$ to $Q_1(1, x_2, x_3, x_4) \equiv$

$Q_2(1, x_2, x_3, x_4) \equiv 0 \pmod{P}$ by finding the roots of the quartics obtained by substituting $w_1 = 1, w_2 = x_2$ into the resultants $\mathrm{Res}_{w_3}(Q_1, Q_2)$ and $\mathrm{Res}_{w_4}(Q_1, Q_2)$. The quartics are solved modulo P. A set of $O(P)$ solutions modulo P is thus determined in time $O(P)$.

Each solution is then lifted to a point $\boldsymbol{x}'$ with $Q_1(\boldsymbol{x}') \equiv Q_2(\boldsymbol{x}') \equiv 0 \pmod{P^2}$ by solving the congruence equations $d_i.[u, v, w] \equiv -P^{-1}Q_i(\boldsymbol{x}) \pmod{P}$, where $d_i = \left[\frac{\partial Q_i(\boldsymbol{x})}{\partial x_2}, \frac{\partial Q_i(\boldsymbol{x})}{\partial x_3}, \frac{\partial Q_i(\boldsymbol{x})}{\partial x_4}\right]$. Solutions to this pair of congruence equations are of the form $[y_1, y_2, y_3] + \lambda[z_1, z_2, z_3]$, and can be scaled so that z_1 is 0 or 1.

Let $\boldsymbol{X} = \boldsymbol{x} + P[0, y_1, y_2, y_3]$ and $\boldsymbol{D} = P[0, z_1, z_2, z_3]$. If $z_1 = 1$ we consider the lattice with basis $\langle \boldsymbol{X}, \boldsymbol{D}, [0, 0, P^2, 0], [0, 0, 0, P^2] \rangle$, while if $z_1 = 0$ the lattice to be used has basis $\langle \boldsymbol{X}, \boldsymbol{D}, [0, P^2, 0, 0], [0, 0, 0, P^2] \rangle$. These lattices have determinant P^5.

Lattice reduction is then used to find short vectors in the lattice, and each vector found is then tested to see whether it is a solution to the equations $Q_1(\boldsymbol{w}) = Q_2(\boldsymbol{w}) = 0$.

Since $P \approx N$, the running time of this method is $O(P) = O(N)$.

Womack implemented his method as a Magma function, and this was incorporated into the V2.11 (2004) release of Magma. The efficiency of the code has since been improved, and it is now included in the internal distribution code of the Magma V2.12 (2005) release [28]. The function is called using the `PointsQI` command.

A similar method, developed independently by myself, takes a complete set of

solutions $\boldsymbol{x}$ modulo P with $x_1 = 1$ and lifts them to a complete set of solutions $\boldsymbol{y} = \boldsymbol{x} + P\boldsymbol{z}$ modulo P^2 by taking the first two terms of the Taylor expansion of $Q_i(\boldsymbol{y}) = Q_i(\boldsymbol{x} + P\boldsymbol{z})$. Each solution $\boldsymbol{y}$ modulo P^2 is then raised to an arbitrary solution $\boldsymbol{y}' = \boldsymbol{y} + P^2\boldsymbol{t}$ modulo P^4. The required solution $\boldsymbol{w}$ can be written as $\boldsymbol{w} = \lambda\boldsymbol{y}' + P^2\boldsymbol{t}$, so solutions lie on a lattice with basis $\langle [1, y_2', y_3', y_4'], P^2[0, u_2, u_3, 1], [0, P^4, 0, 0], [0, 0, P^4, 0] \rangle$ where $\boldsymbol{u} = [0, u_2, u_3, 1]$ is a basis for the solution to the congruence equations $\boldsymbol{t}.\nabla Q_i(\boldsymbol{y}') \equiv 0 \pmod{P^2}$. This lattice has determinant P^{10}.

After lattice reduction, we obtain a new basis $\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4} \rangle$ for the solution $\boldsymbol{w}$, and can write $\boldsymbol{w} = \sum_{i=1}^{4} r_i \boldsymbol{b_i}$, with $r_4 \in \mathbb{Z}$. By the properties of LLL-reduced lattice bases, we have $|r_4| \leq \|\boldsymbol{w}\| \Delta(L)^{-1/4} 2^{3/2}$. Since $\|\boldsymbol{w}\| \leq 2N$ and $\Delta(L) = P^{10}$ we find that by taking $P > 2N^{2/5}$, we have $r_4 = 0$ and the remaining coefficients in the linear combination can be determined. Since Q_1 and Q_2 are homogeneous, we may scale the remaining r_i to be integers.

The running time of this method is $O(P^2) = O(N^{4/5})$. The method has been implemented using Maple (see `p4sim.mws`, appendix D), and results are given in chapter 7.

The `p4sim` method described above can be seen to be the same method as Womack's, but using the modulus P^2 in place of P. In fact, any prime power could be used as the modulus in Womack's method, but there is no benefit in doing so.

To determine a complete set of solutions modulo P^k (where k is a power of 2) by repeatedly lifting a solution modulo P takes time $O(P^k)$. Each of these solutions would then be lifted to an arbitrary solution modulo P^{2k}. The resulting lattice has basis $\langle [1, y_2^{(2k)}, y_3^{(2k)}, y_4^{(2k)}], P^k[0, u_2, u_3, 1], [0, P^{2k}, 0, 0], [0, 0, P^{2k}, 0] \rangle$, and so has

determinant P^{5k}. The size of P is then chosen so that $|r_4| \leq \|\boldsymbol{w}\|\Delta(L)^{-1/4}2^{3/2} < 1$, giving $P > 2^{2/k}N^{4/5k}$. The running time will then be $O(P^k) = O(N^{4/5})$ as before.

However, one improvement which is possible is to lift the complete set of solutions mod P^2 to a set of solutions mod P^6 by taking three terms in the Taylor expansion, rather than just two. This method is described in detail below.

6.3 The `p6sim` method

We first choose a prime $P > 2^{5/6}N^{1/3}$ of good reduction. The prime P should therefore not be a divisor of the discriminants of the two equations. Note that this prime P is not related to the coefficient p of X in the equation $Y^2 = X^3 + pX$ of chapter 4. In the case where the equations are derived via the 2-descent process on an elliptic curve of the form $Y^2 = X^3 + pX$, we may not choose $P = 2$ or $P = p$.

This method uses a complete set of solutions modulo P^2 to find all solutions with $P \nmid w_1$. Note that since we require w_1, w_2, w_3, w_4 to be coprime, we may permute the variables so that $P \nmid w_1$ and then run the algorithm again. It is therefore possible that the algorithm may need to be run four times before all solutions are found. However, if $P \approx 2^{5/6}N^{1/3}$, there are $\frac{1}{2^{5/6}}N^{2/3}$ multiples of P up to N, so the probability that w_1 is one of these multiples is just

$$\frac{\frac{1}{2^{5/6}}N^{2/3}}{N} = \frac{1}{2^{5/6}}N^{-1/3}.$$

Step 1

Find all solutions $\boldsymbol{x} \pmod P$ *with* $x_1 = 1$.

To do this, we first eliminate one variable, say x_4, so that we solve the quartic $F(x_1, x_2, x_3) \equiv 0 \pmod P$.

Since we require $x_1 = 1$, the quartic involves only the variables x_2 and x_3, so we can find a solution in terms of one of these two variables. Then on substituting

each of the possible P values for this variable, we expect to find $O(P)$ solutions $(1, x_2, x_3)$. We then solve the resulting equations to find x_4 in time $O(1)$.

We will therefore find a complete set of $O(P)$ solutions in time $O(P) = O(N^{1/3})$.

Step 2

Lift each solution $\boldsymbol{x}$ *to a set of P solutions* $\boldsymbol{y} \pmod{P^2}$ *where* $\boldsymbol{y} = \boldsymbol{x} + P\boldsymbol{z}$ *for some* $\boldsymbol{z}$, *and* $Q_1(\boldsymbol{y}) \equiv Q_2(\boldsymbol{y}) \equiv 0 \pmod{P^2}$ *with* $y_1 = 1$.

By the Taylor series expansion for vector functions, we have

$$\begin{aligned} Q_i(\boldsymbol{y}) &= Q_i(\boldsymbol{x} + P\boldsymbol{z}) \\ &= Q_i(\boldsymbol{x}) + P\boldsymbol{z}.\nabla Q_i(\boldsymbol{x}) + P^2(\dots) \\ &\equiv Q_i(\boldsymbol{x}) + P\boldsymbol{z}.\nabla Q_i(\boldsymbol{x}) \pmod{P^2}, \end{aligned}$$

where $\nabla Q_i(\boldsymbol{x})$ is the gradient vector of partial derivatives for Q_i evaluated at the point $\boldsymbol{x}$.

So we have $P^{-1}Q_1(\boldsymbol{x}) + \boldsymbol{z}.\nabla Q_1(\boldsymbol{x}) \equiv 0 \pmod{P}$.

The same is true for $Q_2(\boldsymbol{x})$ so we solve the simultaneous congruences

$$P^{-1}Q_1(\boldsymbol{x}) + \boldsymbol{z}.\nabla Q_1(\boldsymbol{x}) \equiv P^{-1}Q_2(\boldsymbol{x}) + \boldsymbol{z}.\nabla Q_2(\boldsymbol{x}) \equiv 0 \pmod{P}. \qquad (6.2)$$

Note that since we require $y_1 = 1$, we must have $z_1 = 0$. These equations are

solved in time $O(1)$.

Since P is a prime of good reduction, P does not divide the discriminants of the two equations (6.1) and so all points modulo P are smooth. This guarantees the existence of a solution to equation (6.2).

Since we have two congruence equations in the three variables z_2, z_3, z_4, we can find a solution in terms of one of these variables, say z_4. Substituting each of the P possible values for z_4, we therefore obtain P solutions $\boldsymbol{z}$ in time $O(P)$ for each $\boldsymbol{x}$. In this way, we have determined a complete set of $O(P^2)$ solutions $\boldsymbol{y}$ modulo P^2 in time $O(P^2) = O(N^{2/3})$.

Step 3

Adjust each solution $\boldsymbol{y}$ *to give a solution* $\boldsymbol{y}'$ $\pmod{P^4}$ *where* $\boldsymbol{y}' = \boldsymbol{y} + P^2\boldsymbol{z}'$ *for some* $\boldsymbol{z}'$, *with* $Q_i(\boldsymbol{y}') \equiv 0 \pmod{P^4}$ *and* $y_1' = 1$.

Using the same method as before, we solve

$$P^{-2}Q_i(\boldsymbol{y}) + \boldsymbol{z}'.\nabla Q_i(\boldsymbol{y}) \equiv 0 \pmod{P^2} \tag{6.3}$$

in time $O(1)$. The existence of a solution to equation (6.3) again follows from the condition that P is a prime of good reduction.

Again, we require $y_1' = 1$ so that $z_1' = 0$. However, this time we require only one solution $\boldsymbol{z}'$ giving just one solution $\boldsymbol{y}'$ for each $\boldsymbol{y}$ and therefore $O(P^2)$ such $\boldsymbol{y}'$.

Step 4

Adjust again to give an arbitrary solution $\boldsymbol{y}''$ (mod P^8) *where* $\boldsymbol{y}'' = \boldsymbol{y}' + P^4\boldsymbol{z}''$ *for some* $\boldsymbol{z}''$, *with* $Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^8}$ *and* $y_1'' = 1$.

By the Taylor series expansion, we have

$$\begin{aligned} Q_i(\boldsymbol{y}'') &= Q_i(\boldsymbol{y}' + P^4\boldsymbol{z}'') \\ &\equiv Q_i(\boldsymbol{y}') + P^4\boldsymbol{z}''.\nabla Q_i(\boldsymbol{y}') \pmod{P^8}. \end{aligned}$$

So we solve the congruences

$$P^{-4}Q_i(\boldsymbol{y}') + \boldsymbol{z}''.\nabla Q_i(\boldsymbol{y}') \equiv 0 \pmod{P^4} \tag{6.4}$$

in time $O(1)$, and the existence of a solution is guaranteed as before.

Step 5

Write the required solution $\boldsymbol{w}$ *as* $\boldsymbol{w} = \lambda\boldsymbol{y}'' + P^2\boldsymbol{t}$ *for some* $\boldsymbol{t}$ *with* $t_1 = 0$.

Since we have found a complete set of solutions $\boldsymbol{y}$ modulo P^2, any solution $\boldsymbol{w} \in \mathbb{Z}^4$ will be projectively equivalent to one of these vectors $\boldsymbol{y}$ modulo P^2. So $\boldsymbol{w} \equiv \lambda\boldsymbol{y}$ (mod P^2).

Now $\boldsymbol{y}' = \boldsymbol{y} + P^2\boldsymbol{z}'$ and $\boldsymbol{y}'' = \boldsymbol{y}' + P^4\boldsymbol{z}''$ so that $\boldsymbol{y} = \boldsymbol{y}'' - P^4\boldsymbol{z}'' - P^2\boldsymbol{z}'$. Then $\boldsymbol{y} \equiv \boldsymbol{y}''$ (mod P^2) and $\boldsymbol{w} \equiv \lambda\boldsymbol{y}''$ (mod P^2). Hence $\boldsymbol{w} = \lambda\boldsymbol{y}'' + P^2\boldsymbol{t}$ for some vector $\boldsymbol{t}$.

Since we require solutions $\boldsymbol{w}$ with $P \nmid w_1$, we have that $P \nmid \lambda$ and $t_1 = 0$.

Step 6

Find a lattice basis for $\boldsymbol{t}$.

By the Taylor series expansion

$$\begin{aligned} Q_i(\boldsymbol{w}) &= Q_i(\lambda \boldsymbol{y}'' + P^2 \boldsymbol{t}) \\ &= Q_i(\lambda \boldsymbol{y}'') + (P^2 \boldsymbol{t}).\nabla Q_i(\lambda \boldsymbol{y}'') + \frac{1}{2}(P^2 \boldsymbol{t})^T [G_i](P^2 \boldsymbol{t}) + \ldots \qquad (6.5) \\ &\equiv \lambda^2 Q_i(\boldsymbol{y}'') + P^2 \lambda \boldsymbol{t}.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^4 \boldsymbol{t}^T [G_i] \boldsymbol{t} \pmod{P^6}, \end{aligned}$$

where $[G_i]$ is the Hessian matrix of second partial derivatives for Q_i. Since Q_i is quadratic, the entries in $[G_i]$ will be constants and will not depend on $\boldsymbol{y}''$.

But $Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^8}$, so $Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^6}$. Hence

$$\begin{aligned} P^2 \lambda \boldsymbol{t}.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^4 \boldsymbol{t}^T [G_i] \boldsymbol{t} &\equiv 0 \pmod{P^6} \\ \Longrightarrow \quad \lambda \boldsymbol{t}.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^2 \boldsymbol{t}^T [G_i] \boldsymbol{t} &\equiv 0 \pmod{P^4}. \end{aligned} \qquad (6.6)$$

Congruence mod P^2 gives $\lambda \boldsymbol{t}.\nabla Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^2}$. But we are searching for solutions with $P \nmid w_1$, so we cannot have $P \mid \lambda$. Hence $\lambda \not\equiv 0 \pmod{P^2}$. Therefore we solve the simultaneous congruences

$$\boldsymbol{t}.\nabla Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^2} \qquad (6.7)$$

in time $O(1)$.

Since $t_1 = 0$, we have two equations in three variables, so the resulting solution space has dimension 1. By solving these equations, we therefore obtain a basis for the solution modulo P^2 consisting of one vector (v_2, v_3, v_4). This can be "standardised" by, for example, multiplying this vector by $v_4^{-1} \pmod{P^2}$, to obtain a vector $(u_2, u_3, 1)$. Note that if v_4 is a multiple of P, we could multiply by the inverse of one of the other elements.

Then since this vector forms a basis for the solution space modulo P^2, we have

$$(t_2, t_3, t_4) \equiv k(u_2, u_3, 1) \pmod{P^2}.$$

We now find a basis for (t_2, t_3, t_4) in $\mathbb{Z}^3$. Suppose that $(t_2, t_3, t_4) = (\alpha, \beta, \gamma) \equiv k(u_2, u_3, 1) \pmod{P^2}$. Then

$$(t_2, t_3, t_4) - \gamma(u_2, u_3, 1) = (\alpha, \beta, \gamma) - (u_2\gamma, u_3\gamma, \gamma) = (\alpha - u_2\gamma, \beta - u_3\gamma, 0)$$

$$\equiv k(u_2, u_3, 1) - \gamma(u_2, u_3, 1) = (k - \gamma)(u_2, u_3, 1) \pmod{P^2}.$$

So we have $(\alpha - u_2\gamma, \beta - u_3\gamma, 0) \equiv (k - \gamma)(u_2, u_3, 1) \pmod{P^2}$. Since the last component is zero, we must have $k - \gamma \equiv 0 \pmod{P^2}$, so that $k \equiv \gamma \pmod{P^2}$.

So now we have $(t_2, t_3, t_4) \equiv \gamma(u_2, u_3, 1) \pmod{P^2}$, so that

$$(t_2, t_3, t_4) - \gamma(u_2, u_3, 1) = (\alpha - u_2\gamma, \beta - u_3\gamma, 0) \equiv 0 \pmod{P^2}.$$

Therefore, $P^2 \mid \alpha - u_2\gamma$ and $P^2 \mid \beta - u_3\gamma$, so that $\alpha - u_2\gamma = P^2 a$ and $\beta - u_3\gamma = P^2 b$ where $a, b \in \mathbb{Z}$. Then we have

$$(t_2, t_3, t_4) = \gamma(u_2, u_3, 1) + a(P^2, 0, 0) + b(0, P^2, 0),$$

so that a basis for (t_2, t_3, t_4) is given by

$$\langle (u_2, u_3, 1), (P^2, 0, 0), (0, P^2, 0) \rangle.$$

We can then write

$$\begin{aligned} \boldsymbol{t} = (t_1, t_2, t_3, t_4) &= (0, t_2, t_3, t_4) \\ &= \gamma(0, u_2, u_3, 1) + a(0, P^2, 0, 0) + b(0, 0, P^2, 0) \\ &= \gamma(0, u_2, u_3, 1) + P^2(0, a, b, 0) \\ &= \gamma \boldsymbol{u} + P^2 \boldsymbol{m}. \end{aligned}$$

Step 7

Adjust $\boldsymbol{u}$ *to give* $\boldsymbol{u}'$ *such that* $P^4 \mid \boldsymbol{u}'.\nabla Q_i(\boldsymbol{y}'')$, *with* $\boldsymbol{u}' = \boldsymbol{u} + P^2\boldsymbol{v}$ *for some* $\boldsymbol{v}$.

We have $\boldsymbol{u}'.\nabla Q_i(\boldsymbol{y}'') = (\boldsymbol{u} + P^2\boldsymbol{v}).\nabla Q_i(\boldsymbol{y}'') = \boldsymbol{u}.\nabla Q_i(\boldsymbol{y}'') + P^2\boldsymbol{v}.\nabla Q_i(\boldsymbol{y}'')$.

So we solve the simultaneous congruences $\boldsymbol{u}.\nabla Q_i(\boldsymbol{y}'') + P^2\boldsymbol{v}.\nabla Q_i(\boldsymbol{y}'') \equiv 0 \pmod{P^4}$ to find $\boldsymbol{v}$ in time $O(1)$, and then determine $\boldsymbol{u}'$. Note that $\boldsymbol{u}'$ will be of the form $(0, u_2', u_3', 1)$.

Then

$$\boldsymbol{t} = \gamma \boldsymbol{u}' + P^2 \boldsymbol{m}' \tag{6.8}$$

for some vector $\boldsymbol{m}' = (0, m_2, m_3, 0)$.

Step 8

Determine $\boldsymbol{m}'$.

Equation (6.6) becomes

$$\lambda(\gamma \boldsymbol{u}' + P^2 \boldsymbol{m}').\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^2 (\gamma \boldsymbol{u}' + P^2 \boldsymbol{m}')^T [G_i](\lambda \boldsymbol{u}' + P^2 \boldsymbol{m}') \equiv 0 \pmod{P^4}$$

$$\implies \lambda\gamma \boldsymbol{u}'.\nabla Q_i(\boldsymbol{y}'') + \lambda P^2 \boldsymbol{m}'.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^2 \gamma^2 \boldsymbol{u}'^T [G_i] \boldsymbol{u}' \equiv 0 \pmod{P^4}.$$

But $P^4 \mid \boldsymbol{u}'.\nabla Q_i(\boldsymbol{y}'')$, so

$$\lambda P^2 \boldsymbol{m}'.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} P^2 \gamma^2 \boldsymbol{u}'^T [G_i] \boldsymbol{u}' \equiv 0 \pmod{P^4}$$

$$\implies \lambda \boldsymbol{m}'.\nabla Q_i(\boldsymbol{y}'') + \frac{1}{2} \gamma^2 \boldsymbol{u}'^T [G_i] \boldsymbol{u}' \equiv 0 \pmod{P^2}.$$

Let $\boldsymbol{u}'^T [G_1] \boldsymbol{u}' = r_1$, and $\boldsymbol{u}'^T [G_2] \boldsymbol{u}' = r_2$.

Then $\lambda \boldsymbol{m}'.\nabla Q_1(\boldsymbol{y}'') + \frac{1}{2}\gamma^2 r_1 \equiv 0 \pmod{P^2}$ and $\lambda \boldsymbol{m}'.\nabla Q_2(\boldsymbol{y}'') + \frac{1}{2}\gamma^2 r_2 \equiv 0 \pmod{P^2}$,

so that

$$\lambda r_2 \boldsymbol{m}'.\nabla Q_1(\boldsymbol{y}'') - \lambda r_1 \boldsymbol{m}'.\nabla Q_2(\boldsymbol{y}'') \equiv 0 \pmod{P^2},$$

or

$$\lambda[r_2\boldsymbol{m}'.\nabla Q_1(\boldsymbol{y}'') - r_1\boldsymbol{m}'.\nabla Q_2(\boldsymbol{y}'')] \equiv 0 \pmod{P^2}.$$

But $P \nmid \lambda$, so $r_2\boldsymbol{m}'.\nabla Q_1(\boldsymbol{y}'') - r_1\boldsymbol{m}'.\nabla Q_2(\boldsymbol{y}'') \equiv 0 \pmod{P^2}$.

We have $\boldsymbol{m}' = (0, m_2, m_3, 0)$, so

$$r_2(0, m_2, m_3, 0).\nabla Q_1(\boldsymbol{y}'') - r_1(0, m_2, m_3, 0).\nabla Q_2(\boldsymbol{y}'') \equiv 0 \pmod{P^2}$$

$$\implies r_2 m_2 \frac{\partial Q_1}{\partial x_2}(\boldsymbol{y}'') + r_2 m_3 \frac{\partial Q_1}{\partial x_3}(\boldsymbol{y}'') - r_1 m_2 \frac{\partial Q_2}{\partial x2}(\boldsymbol{y}'') - r_1 m_3 \frac{\partial Q_2}{\partial x_3}(\boldsymbol{y}'') \equiv 0 \pmod{P^2}$$

$$= nP^2$$

for some $n \in \mathbb{Z}$.

This gives

$$r_2 m_3 \frac{\partial Q_1}{\partial x_3}(\boldsymbol{y}'') - r_1 m_3 \frac{\partial Q_2}{\partial x_3}(\boldsymbol{y}'') = nP^2 + r_1 m_2 \frac{\partial Q_2}{\partial x_2}(\boldsymbol{y}'') - r_2 m_2 \frac{\partial Q_1}{\partial x_2}(\boldsymbol{y}'')$$

$$\implies m_3 \left[r_2 \frac{\partial Q_1}{\partial x_3}(\boldsymbol{y}'') - r_1 \frac{\partial Q_2}{\partial x_3}(\boldsymbol{y}'') \right] = nP^2 + m_2 \left[r_1 \frac{\partial Q_2}{\partial x_2}(\boldsymbol{y}'') - r_2 \frac{\partial Q_1}{\partial x_2}(\boldsymbol{y}'') \right]$$

$$\implies Am_3 = nP^2 + Bm_2$$

$$\implies m_3 \equiv c_1 m_2 \pmod{P^2}.$$

Then

$$\boldsymbol{m}' = (0, m_2, m_3, 0) = m_2(0, 1, c_1, 0) + c_2(0, 0, P^2, 0). \tag{6.9}$$

Step 9

Find a lattice basis for the solution $\boldsymbol{w}$. *Use lattice reduction techniques in order to make this the "best possible".*

So $t = (0, t_2, t_3, t_4) = \gamma \boldsymbol{u}' + P^2 \boldsymbol{m}' = \gamma(0, u_2', u_3', 1) + P^2(0, m_2, m_3, 0)$, with $m_3 \equiv c_1 m_2 \pmod{P^2}$.

Then we have

$$\begin{aligned} \boldsymbol{w} &= \lambda \boldsymbol{y}'' + P^2 \boldsymbol{t} \\ &= \lambda \boldsymbol{y}'' + P^2(\gamma \boldsymbol{u}' + P^2 \boldsymbol{m}') \\ &= \lambda \boldsymbol{y}'' + \gamma P^2 \boldsymbol{u}' + P^4 \boldsymbol{m}' \\ &= \lambda \boldsymbol{y}'' + \gamma P^2 \boldsymbol{u}' + P^4(m_2(0, 1, c_1, 0) + c_2(0, 0, P^2, 0)) \\ &= \lambda \boldsymbol{y}'' + \gamma P^2 \boldsymbol{u}' + m_2(0, P^4, P^4 c_1, 0) + c_2(0, 0, P^6, 0). \end{aligned}$$

This has basis

$$\langle \boldsymbol{b}_1^*, \boldsymbol{b}_2^*, \boldsymbol{b}_3^*, \boldsymbol{b}_4^* \rangle = \langle \boldsymbol{y}'', (0, P^2 u_2', P^2 u_3', P^2), (0, P^4, P^4 c_1 0), (0, 0, P^6, 0) \rangle. \qquad (6.10)$$

The lattice has determinant P^{12}, since

$$\begin{vmatrix} 1 & y_2'' & y_3'' & y_4'' \\ 0 & u_2' P^2 & u_3' P^2 & P^2 \\ 0 & P^4 & P^4 c_1 & 0 \\ 0 & 0 & P^6 & 0 \end{vmatrix} = \begin{vmatrix} u_2' P^2 & u_3' P^2 & P^2 \\ P^4 & P^4 c_1 & 0 \\ 0 & P^6 & 0 \end{vmatrix} = P^2 P^4 P^6 = P^{12}.$$

We can then use the LLL-reduction algorithm on this basis to produce a new basis $\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4} \rangle$.

Step 10

Express $\boldsymbol{w}$ in terms of the lattice basis vectors.

Writing $\boldsymbol{w}$ as a linear combination of the lattice basis vectors, we have

$$\boldsymbol{w} = \sum_{i=1}^{4} r_i^* \boldsymbol{b}_i^* = \sum_{i=1}^{4} r_i \boldsymbol{b}_i,$$

with $r_4 \in \mathbb{Z}$.

Then

$$\boldsymbol{b_1} = \boldsymbol{b_1^*}$$

$$\boldsymbol{b_2} = \mu_{2,1}\boldsymbol{b_1^*} + \boldsymbol{b_2^*}$$

$$\boldsymbol{b_3} = \mu_{3,1}\boldsymbol{b_1^*} + \mu_{3,2}\boldsymbol{b_2^*} + \boldsymbol{b_3^*}$$

$$\boldsymbol{b_4} = \mu_{4,1}\boldsymbol{b_1^*} + \mu_{4,2}\boldsymbol{b_2^*} + \mu_{4,3}\boldsymbol{b_3^*} + \boldsymbol{b_4^*}.$$

So

$$\boldsymbol{w} = r_1\boldsymbol{b_1^*} + r_2(\mu_{2,1}\boldsymbol{b_1^*} + \boldsymbol{b_2^*}) + r_3(\mu_{3,1}\boldsymbol{b_1^*} + \mu_{3,2}\boldsymbol{b_2^*} + \boldsymbol{b_3^*}) + r_4(\mu_{4,1}\boldsymbol{b_1^*} + \mu_{4,2}\boldsymbol{b_2^*} + \mu_{4,3}\boldsymbol{b_3^*} + \boldsymbol{b_4^*})$$

$$= (r_1 + r_2\mu_{2,1} + r_3\mu_{3,1} + r_4\mu_{4,1})\boldsymbol{b_1^*} + (r_2 + r_3\mu_{3,2} + r_4\mu_{4,2})\boldsymbol{b_2^*} + (r_3 + r_4\mu_{4,3})\boldsymbol{b_3^*} + r_4\boldsymbol{b_4^*}.$$

Hence $r_4^* = r_4$.

Now

$$\|\boldsymbol{w}\|^2 = \sum_{i=1}^{4} \|r_i^* \boldsymbol{b}_i^*\|^2 = \sum_{i=1}^{4} |r_i^*|^2 \|\boldsymbol{b}_i^*\|^2 \geq |r_4|^2 \|\boldsymbol{b}_4^*\|^2. \tag{6.11}$$

By theorem 5.2.2(a), we have

$$\|\boldsymbol{b}_i\|^2 \leq 2^3 \|\boldsymbol{b}_4^*\|^2.$$

Then, using theorem 5.2.2(b), we have

$$\Delta(L)^2 \leq \prod_{i=1}^{4} \|\boldsymbol{b}_i\|^2 \leq 2^{12} \|\boldsymbol{b}_4^*\|^8.$$

This implies that $\|\boldsymbol{b}_4^*\|^2 \geq \Delta(L)^{1/2} 2^{-3}$.

By (6.11) we find that

$$\begin{aligned} &\|\boldsymbol{w}\|^2 \geq |r_4|^2 \|\boldsymbol{b}_4^*\|^2 \geq |r_4|^2 \Delta(L)^{1/2} 2^{-3} \\ \Longrightarrow\ &|r_4|^2 \leq \|\boldsymbol{w}\|^2 \Delta(L)^{-1/2} 2^3 \qquad (6.12) \\ \Longrightarrow\ &|r_4| \leq \|\boldsymbol{w}\| \Delta(L)^{-1/4} 2^{3/2}. \end{aligned}$$

We are searching for solutions $\boldsymbol{w}$ with $|w_i| \leq N$. So we have $\|\boldsymbol{w}\|^2 \leq 4N^2 \implies \|\boldsymbol{w}\| \leq 2N$. We also know that $\Delta(L) = P^{12}$.

Then (6.12) becomes

$$|r_4| \leq 2^{5/2} P^{-3} N.$$

However, $P > 2^{5/6}N^{1/3}$, so $P^3 > 2^{5/2}N$. Hence $P^{-3} < 2^{-5/2}N^{-1}$. So we now have

$$|r_4| < 2^{5/2}(2^{-5/2}N^{-1})N = 1.$$

Since r_4 is an integer, we must have $r_4 = 0$. Thus we can write

$$\boldsymbol{w} = r_1\boldsymbol{b_1} + r_2\boldsymbol{b_2} + r_3\boldsymbol{b_3} \tag{6.13}$$

for some integral r_1, r_2, r_3.

Step 11

Find r_1, r_2 and r_3.

On substituting (6.13) into (6.1), we obtain a pair of simultaneous quadratic equations in three variables, r_1, r_2 and r_3.

Eliminating one of the variables, say r_3, gives a homogeneous quartic equation in the two remaining variables r_1 and r_2. We can then divide through by one of the remaining variables, say r_2, to obtain a quartic polynomial f in a rational variable x for which we require the solution r_1/r_2,

$$f(x) = ax^4 + bx^3 + cx^2 + dx + e = 0,$$

where $a, b, c, d, e \in \mathbb{Z}$.

If $f(x)$ has a linear factor $(r_2x - r_1)$, so that $x = r_1/r_2$ is a root, then by the rational root theorem we have $r_2 \mid a$, and $r_1 \mid e$. Then $ax = ar_1/r_2 \in \mathbb{Z}$ since $r_2 \mid a$.

Using Newton's method (or some other numeric method), we can compute roots x_0 to within an accuracy of $1/a$. Multiplying these roots by a, we have therefore computed ax_0 to within 1. Since $ax \in \mathbb{Z}$, we take the integer part of the computed ax_0 to give an exact value for $ax = ar_1/r_2$. Dividing by a gives us r_1/r_2, which is rational.

So $r_1/r_2 = [ax_0]/a$ and we then cancel any common factors in the numerator and denominator to give r_1/r_2 in lowest terms.

Having found r_1 and r_2, the remaining variable r_3 can be found in time $O(1)$.

The running time of this method is $O(P^2) = O(N^{2/3})$.

Note that the two equations $Q_i = 0$ are required to be homogeneous so that we can take λ as a common factor in the Taylor series expansion (6.5). Since the equations are quadratic, we can take a maximum of three terms in the Taylor series expansion for $Q_i(\boldsymbol{w}) = Q_i(\lambda \boldsymbol{y}'' + P^2 \boldsymbol{t})$ since all subsequent terms will be zero. The method can easily be parallelised if required by breaking the outer loop into several pieces.

6.4 The p4quart method

We may also choose to work with the ternary quartic which arises from eliminating one of the variables from the pair of simultaneous equations. Womack [53] developed a method for finding points on the quadric intersection by searching for points on a homogeneous ternary polynomial. This involved a sieving and filtering technique, and worked simultaneously on the three polynomials $F_1 = F(x, y, z)$, $F_2 = F(x, z, y)$ and $F_3 = F(z, y, x)$ in order to ensure that a point on F was found in time dependent on the second largest coordinate of the solution. Womack was concerned with finding a solution to the quadric intersection, and later abandoned this in favour of the p-adic Elkies method for solving the pair of simultaneous equations directly.

However, it is possible to adapt the p-adic Elkies method to solve the quartic, as described below. The method will work on a smooth plane curve of any degree, but it is presented here to solve the ternary quartic.

One variable, say x_4 is first eliminated from the pair of simultaneous equations (6.1) to leave a homogeneous ternary quartic equation $F(\boldsymbol{w}) = 0$, for which we seek solutions $\boldsymbol{w} \in \mathbb{Z}^3$ in the region $|w_i| \leq N$.

We choose a prime $P > 12^{1/4}N^{1/2}$, and determine a complete set of $O(P)$ smooth solutions $\boldsymbol{x}$ modulo P with $x_1 = 1$ in time $O(P)$.

Each of these smooth solutions can then be raised to a set of $O(P)$ solutions $\boldsymbol{y}$ modulo P^2, where $\boldsymbol{y} = \boldsymbol{x} + P\boldsymbol{z}$ and $y_1 = 1$. Since this step takes time $O(P)$, the set

of solutions modulo P^2 is found in time $O(P^2)$.

Each solution $\boldsymbol{y}$ modulo P^2 is then raised to an arbitrary solution $\boldsymbol{y}'$ modulo P^4, where $\boldsymbol{y}' = \boldsymbol{y} + P^2\boldsymbol{z}'$ and $y'_1 = 1$, using a Taylor series expansion.

The required solution $\boldsymbol{w}$ then lies on a lattice with basis $\langle \boldsymbol{y}', P^2(0, u_2, 1), P^2(0, P^2, 0)\rangle$, where $\boldsymbol{u} = [0, u_2, 1]$ is a basis for the solution to the congruence equation $\boldsymbol{t}.\nabla F(\boldsymbol{y}') \equiv 0 \pmod{P^2}$. This lattice has determinant P^6.

We use lattice reduction on this basis to produce a new basis $\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}\rangle$ for the solution $\boldsymbol{w}$, and can write $\boldsymbol{w} = \sum_{i=1}^{3} r_i \boldsymbol{b_i}$, with $r_3 \in \mathbb{Z}$. Again, we can eliminate the final basis vector, since $|r_3| \leq 2\|\boldsymbol{w}\|\Delta(L)^{-1/3} = 2\sqrt{3}NP^{-2} < 1$ so that we must have $r_3 = 0$. Then $\boldsymbol{w}$ can be written as a linear combination of the first two basis vectors, and integer coefficients r_1 and r_2 can be determined in time $O(1)$.

The running time of this method is $O(P^2) = O(N)$.

In the case where we have only one equation, we cannot take more than two terms in the Taylor series expansion, since we needed two equations in order to eliminate γ in step 8 of `p6sim`.

6.5 The `composite` method

We have already seen that we can use a prime power as a modulus in Womack's p-adic version of the Elkies method (see section 6.2), but in fact we can use any composite number M as the modulus.

Suppose $N(M)$ is the number of vectors $\boldsymbol{w} \in \mathbb{Z}^4$ for which $Q_i(\boldsymbol{w}) \equiv 0 \pmod{M}$. We expect $N(M) \approx M$.

Let $\rho(M) = \frac{N(M)}{M}$. For a fixed prime p, we expect that $\rho(p^e)$ is eventually constant as $e \to \infty$. We also expect that $\rho(uv) \approx \rho(u)\rho(v)$.

Then if M is a product of prime powers, $p_i^{e_i}$, with $\rho\left(p_i^{e_i}\right) < 1$, we expect $\rho(M)$ to be small. This means that there will be a relatively small number of solutions modulo M. These solutions can be found by considering each prime power separately.

We first compute the number $q_0 = 2^{5/3}N^{2/3}$, and build a composite number $q = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$, where $r \geq 2$ and $\alpha_i \geq 1$, such that $q > q_0$.

It should be noted that it is possible for solutions to exist such that each component is divisible by one of the prime factors of q. Such solutions will not be found using this method.

We then attempt to find a complete set of (projectively different) solutions $\boldsymbol{x}$ modulo q. To do this, we first find a set of (projectively different) solutions $\boldsymbol{x}$ modulo $p_i^{\alpha_i}$ for each i such that $1 \leq i \leq r$. Here, taking the first non-zero component to be 1 ensures the solutions are projectively different. The solutions will therefore be

of one of the forms $(1, x_2, x_3, x_4)$, $(0, 1, x_3, x_4)$, $(0, 0, 1, x_4)$ or $(0, 0, 0, 1)$. There will be $O\left(p_i^{\alpha_i}\right)$ solutions for each i, and these can be found in time $O\left(p_i^{\alpha_i}\right)$ using the method detailed for step 1 of the `p6sim` method (section 6.3). Then each solution modulo $p_i^{\alpha_i}$ is combined with each combination of the solutions modulo $p_j^{\alpha_j}$ with $j \neq i$ using the Chinese remainder theorem.

Theorem 6.5.1 (Chinese Remainder Theorem). *Suppose that*

$$\begin{aligned} Q_1(\boldsymbol{x_1}) \equiv Q_2(\boldsymbol{x_1}) &\equiv 0 \pmod{p_1^{\alpha_1}} \\ Q_1(\boldsymbol{x_2}) \equiv Q_2(\boldsymbol{x_2}) &\equiv 0 \pmod{p_2^{\alpha_2}} \\ &\vdots \\ Q_1(\boldsymbol{x_r}) \equiv Q_2(\boldsymbol{x_r}) &\equiv 0 \pmod{p_r^{\alpha_r}}, \end{aligned}$$

where $q = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}$. *Let* $M_i = \frac{q}{p_i^{\alpha_i}}$. *Then there exist integers* n_i *such that* $n_i M_i \equiv 1 \pmod{p_i^{\alpha_i}}$. *A solution* $\boldsymbol{x}$ *modulo* q *is then given by*

$$\boldsymbol{x} = n_1 M_1 \boldsymbol{x_1} + n_2 M_2 \boldsymbol{x_2} + \ldots + n_r M_r \boldsymbol{x_r}.$$

The Chinese remainder theorem takes time $O(1)$ and would need to be performed $O(q) = O\left(p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_r^{\alpha_r}\right)$ times.

So a set of $O(q)$ projectively different solutions mod q is found in time $O(q)$.

Each solution $\boldsymbol{x}$ modulo q is then lifted twice to produce an arbitrary solution $\boldsymbol{y'}$ modulo q^4 using Taylor series expansions as in `p6sim`, and then reduced to give a solution $\boldsymbol{y''}$ modulo q^3.

The required solution $\boldsymbol{w}$ can be written as $\boldsymbol{w} = \lambda y'' + q\boldsymbol{t}$ for some vector $\boldsymbol{t}$, so solutions lie on a lattice with basis $\langle \boldsymbol{y}'', [0, qu_2, qu_3, q], [0, q^2, q^2c_1, 0], [0, 0, q^3, 0] \rangle$. The method for determining the vector $\boldsymbol{u} = [0, u_2, u_3, 1]$ and the constant c_1 follows steps 6 to 8 of `p6sim`. The lattice has determinant q^6.

After lattice reduction, we obtain a new basis $\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4} \rangle$ for the solution $\boldsymbol{w}$, and can write $\boldsymbol{w} = \sum_{i=1}^{4} r_i \boldsymbol{b_i}$, with $r_4 \in \mathbb{Z}$. By the properties of LLL-reduced lattice bases, we have $|r_4| \leq \|\boldsymbol{w}\| \Delta(L)^{-1/4} 2^{3/2} \leq 2^{5/2} N q^{-3/2}$. But $q > 2^{5/3} N^{2/3}$, so we have $|r_4| < 1$ as before. The remaining constants in the linear combination can then be determined in time $O(1)$.

The running time of this method is $O(q) = O(N^{2/3})$, which appears to give no time saving over `p6sim`. However, by choosing the $p_i^{\alpha_i}$ carefully, we can construct a q for which the number of solutions modulo q is much smaller than before.

6.6 Maple Worksheets

Maple worksheets to perform `p6sim`, `p4quart`, `p4sim` and `composite` have been included in Appendices B, C, D and E respectively. The worksheets read in the equations file previously created by the descent worksheet (see Appendix A). Where possible, standard Maple commands have been used (for example, Maple has a built-in LLL reduction algorithm).

Chapter 7

Results and Conclusions

7.1 Overview

The maple worksheets for the descent and search algorithms have been run using Maple 7 on a PC with an Intel Pentium 4, 1.8GHz processor, and 512MB memory. Timings were taken for a selection of equations of the form $Y^2 = X^3 + pX$ where $p \equiv 5 \pmod 8$.

The expected rank and height of a generator on each curve, according to the Birch and Swinnerton-Dyer conjecture, was first computed using the `ellanalyticrank` command from Cremona's `heegner.gp` script [15]. In all cases, the expected rank was confirmed to be 1. The heights (to 4 decimal places) are presented in table 7.1.

Prime	$L'(E,1)$	height	time (secs)
317	22.2386	50.6111	2.094
461	16.7067	41.7531	3.063
509	9.0854	23.2754	3.375
1213	7.4786	23.8043	7.859
1637	19.1955	65.8544	10.375
2917	25.8298	102.3829	18.719
3389	8.2425	33.9196	21.687
4861	18.4057	82.8910	31.032
5701	11.7432	55.0360	35.984
6229	18.1358	86.8988	39.141
6829	12.0335	59.0004	42.828
6869	10.3954	51.0432	43.125
7333	5.9365	29.6294	47.250
7349	14.8152	73.9841	46.609
7901	19.7661	100.5113	50.609
8221	22.1941	113.9832	53.891
8269	7.5221	38.6879	52.516
8293	10.9213	56.2115	52.718
8501	17.6933	91.6325	53.657
8941	17.5949	92.2794	57.156
12517	8.4282	48.0819	103.047
16421	22.4762	137.2290	108.672
17293	3.2505	20.1044	109.484
17509	5.5336	34.3320	78.984
17789	13.9961	87.1801	110.860

Table 7.1: Expected heights of generators

A generator was then found on each of the curves. We illustrate with a fully worked example. Consider the case $p = 2917$.

For $p = 2917$, we have $u = 1$, $v = -27$. Then

$$b^2 = 27m^2 - 27l^2 + lm.$$

This is equivalent to the equation

$$-LM + 2917N^2 = 0,$$

where

$$L = 2b + 53l - 55m$$

$$M = 1458b + 37205l - 38644m \tag{7.1}$$

$$N = b + 26l - 27m.$$

The sign of b can be chosen so that $2917 \nmid M$, so that

$$L = \pm 2917\theta^2$$

$$M = \pm\phi^2 \tag{7.2}$$

$$N = \pm\theta\phi$$

for some integers θ and ϕ.

On solving (7.1) for b, l, m and using (7.2), we have

$$\begin{aligned} \pm b &= -2126493\theta^2 + 2917\theta\phi - \phi^2 \\ \pm l &= -2047734\theta^2 + 2862\theta\phi - \phi^2 \\ \pm m &= -2050651\theta^2 + 2864\theta\phi - \phi^2. \end{aligned} \tag{7.3}$$

Now $b \equiv 1 \pmod 2$, and (4.18) implies

$$\begin{aligned} l &\equiv 3 \pmod 4 \\ m &\equiv 0 \pmod 4. \end{aligned} \tag{7.4}$$

A check on all possibilities modulo 4 reveals that the upper signs in (7.3) are incompatible with (7.4).

The equations can be simplified by letting $\psi = \phi - 1431\theta$, then

$$\begin{aligned} b &= -27\theta^2 + 55\theta\psi - \psi^2 \\ l &= 27\theta^2 - \psi^2 \\ m &= -28\theta^2 + 2\theta\psi - \psi^2. \end{aligned} \tag{7.5}$$

Letting $l + im = (c + id)^2 = \gamma^2$, say, and taking the lower sign in (7.5), we obtain from equation (4.20)

$$\gamma^2 = (27 - 28i)\theta^2 + 2i\theta\psi + (-1 - i)\psi^2.$$

This is equivalent to the equation

$$-RT + (54 - i)S^2 = 0, \tag{7.6}$$

where

$$\begin{aligned} R &= (-1 - 6i)\gamma + (-3 - 4i)\psi + (27 + 30i)\theta \\ T &= (-39 + 21i)\gamma + (-25 + 35i)\psi + (133 - 277i)\theta \\ S &= (-2 - i)\gamma - 2\psi + (15 - 2i)\theta. \end{aligned} \tag{7.7}$$

Solving (7.7) for γ, ψ, θ, we have

$$\begin{aligned} \gamma &= \left(\frac{39}{2} - \frac{21}{2}i\right)R + \left(\frac{1}{2} + 3i\right)T + (-109 - 52i)\,S \\ \psi &= -14iR + \left(\frac{3}{2} + \frac{1}{2}i\right)T + (-56 + 41i)\,S \\ \theta &= \left(\frac{7}{2} - \frac{3}{2}i\right)R + \frac{1}{2}iT + (-17 - 11i)\,S, \end{aligned} \tag{7.8}$$

so that

$$\begin{aligned} 2\psi &= -28iR + (3 + i)T + (-112 + 82i)S \\ 2\theta &= (7 - 3i)R + iT + (-34 - 22i)S. \end{aligned} \tag{7.9}$$

Now

$$\begin{aligned} b &\equiv \theta^2 + \theta\psi + \psi^2 \equiv 1 \pmod 2 \\ l &\equiv \theta^2 + \psi^2 \equiv 1 \pmod 2 \\ m &\equiv \psi^2 \equiv 0 \pmod 2, \end{aligned}$$

so we must have

$$\begin{aligned} \theta &\equiv 1 \pmod 2 \\ \psi &\equiv 0 \pmod 2. \end{aligned} \tag{7.10}$$

The sign of γ can be chosen so that $(54 - i) \nmid R$, so that by (7.6) we must have $(54 - i) \mid T$.

Then there are four cases to consider, where $\lambda = x + iy$ and $\mu = w + iz$ are Gaussian integers:

1. $R = \lambda^2, \quad T = (54 - i)\mu^2, \quad S = \lambda\mu$

2. $R = (1 + i)\lambda^2, \quad T = (1 + i)(54 - i)\mu^2, \quad S = (1 + i)\lambda\mu$

3. $R = i\lambda^2, \quad T = i(54 - i)\mu^2, \quad S = i\lambda\mu$

4. $R = i(1 + i)\lambda^2, \quad T = i(1 + i)(54 - i)\mu^2, \quad S = i(1 + i)\lambda\mu$

Substituting each of these four cases into the equations for 2θ and 2ψ (7.9), and knowing that $\theta \equiv 1 \pmod 2$ and $\psi \equiv 0 \pmod 2$ from equation (7.10), a search of all possibilities for (x, y, w, z) modulo 16 reveals that cases 1 and 3 are impossible.

We know that θ and ψ are (rational) integers, so their imaginary parts must be

zero. Taking the imaginary parts from case 2, we therefore have

$$\begin{aligned}\Im(2\theta) = {} & 20xy + 4x^2 - 4y^2 - 12xz - 12yw - 56xw + 56yz - 106wz \\ & + 55w^2 - 55z^2 = 0 \\ \Im(2\psi) = {} & -194xz - 194yw - 30xw + 30yz + 224wz + 214w^2 - 214z^2 \\ & + 56xy - 28x^2 + 28y^2 = 0\end{aligned} \tag{7.11}$$

and these equations must be solved simultaneously.

The maple worksheet descent.mws (see appendix A) found these equations in 3.875 seconds. The worksheet `p6sim.mws` (see appendix B) was then used to find a solution to the simultaneous equations.

From table 7.1, the canonical height of the generator of this curve is 102.3829, so solutions $\boldsymbol{w}$ are expected with $|w_i| \leq 602$. The prime chosen for the p-adic lifting is $P = 17$. On eliminating the variable z and setting $x = 1$, solutions modulo P are sought to the equation

$$\begin{aligned}F = {} & 12yw + 16y + 11w + 2y^2 + 2w^2 + 3y^2w^2 + 9yw^2 \\ & + 2y^3w + 16w^3y + 8y^3 + 12y^4 + 13w^3 + 2w^4 + 2\end{aligned}$$

The corresponding value of the variable z is then found. Eight solutions $\boldsymbol{x}$ to the simultaneous equations (7.11) are found, including $\boldsymbol{x_0} = (1, 7, 16, 4)$.

On substituting this solution $\boldsymbol{x_0}$ into equation (6.2) and solving for $\boldsymbol{z}$ modulo P, we obtain a set of seventeen solutions, including $\boldsymbol{z_0} = (0, 10, 8, 0)$. A solution modulo P^2 to the simultaneous equations is then given by $\boldsymbol{y_0} = (1, 177, 152, 4)$.

The vector $\boldsymbol{y_0}$ is then raised to an arbitrary solution modulo P^4. On substituting $\boldsymbol{y_0}$ into equation (6.3) and solving for $\boldsymbol{z'}$, we find the solution $\boldsymbol{z'_0} = (0, 274, 64, 1)$, which gives the solution $\boldsymbol{y'_0} = (1, 79363, 18648, 293)$ to the equations modulo P^4.

This is then raised to an arbitrary solution modulo P^8. On substituting $\boldsymbol{y'_0}$ into equation (6.4) and solving for $\boldsymbol{z''}$, we find the solution $\boldsymbol{z''_0} = (0, 94668, 66522, 1)$, so that $\boldsymbol{y''_0} = (1, 931087950, 5556002610, 83814)$ is a solution to the equations modulo P^8.

When this is substituted into equation (6.7) and solved for $\boldsymbol{t}$, the standardised basis vector $\boldsymbol{u} = (0, u_2, u_3, 1) = (0, 49513, 51548, 1)$ is found.

We then find a basis for the vector $\boldsymbol{m'}$ in equation (6.8). As in equation (6.9), we find that $\boldsymbol{m'} = m_2(0, 1, 22, 0) + c_2(0, 0, P^2, 0)$.

A basis for the solution $\boldsymbol{w}$ to the simultaneous equations is therefore given by

$$\langle \boldsymbol{y''}, P^2(0, u_2, u_3, 1), P^4(0, 1, c_1, 0), P^4(0, 0, P^2, 0)\rangle$$

$$= \langle (1, 931087950, 5556002610, 83814), (0, 14309257, 14897372, 289),$$

$$(0, 1837462, 83521, 0), (0, 0, 24137569, 0)\rangle.$$

After this basis is LLL-reduced, and ignoring the fourth basis vector, we are left with

$$\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}\rangle = \langle (469, 648, 7997, 5344), (1538, 6346, 1131, -1940),$$

$$(-5907, 352, -1386, 1804)\rangle.$$

These vectors are substituted into equation (6.13), and the constants are found to be $r_1 = r_2 = 0, r_3 = 1$.

Therefore a solution $\boldsymbol{w}$ is given by

$$\begin{aligned}\boldsymbol{w} &= r_1\boldsymbol{b_1} + r_2\boldsymbol{b_2} + r_3\boldsymbol{b_3} \\ &= 0(469, 648, 7997, 5344) + 0(1538, 6346, 1131, -1940) \\ &\qquad + 1(-5907, 352, -1386, 1804) \\ &= (-5907, 352, -1386, 1804).\end{aligned}$$

Since the simultaneous equations are homogeneous, we may divide through by any common divisors to find a coprime solution. Since $\gcd(-5907, 352, -1386, 1804) = 11$, we find the coprime solution $\boldsymbol{w} = (-537, 32, -126, 164)$.

This solution was found by `p6sim.mws` in 9.063 seconds, by `p4quart.mws` (see appendix C) in 285.545 seconds, and by `p4sim.mws` (see appendix D) in 17.752 seconds.

So we have

$$\lambda = -32 - 537i$$

$$\mu = -164 - 126i$$

and

$$R = (1 + i)\lambda^2$$

$$T = (1 + i)(54 - i)\mu^2$$

$$S = (1 + i)\lambda\mu,$$

so that by equations (7.8) and (7.5),

$$b = 76374752761$$

$$l = 2107398267$$

$$m = -14891522644.$$

From equation (4.18), we deduce that

$$c = -92594$$

$$d = 80413,$$

and $a = c^2 + d^2$, so that

$$a = 15039899405.$$

Finally, we have $r = (u + 2iv)(c + id)^4 - 2ib$, $s = 2ab$ and $t = pa^4 + 4b^4$, so that

$$r = -3606612172600350805831$$

$$s = 2297938793190572014410$$

$$t = 285493596261203315374167041270080026045845689.$$

Substituting these values into equation (4.4), we obtain the generator (x, y) with

coordinates

$$x = \frac{1300765136354902263192060743992593106360056 1}{5280522697250142498582105706936285247648100},$$

$$y = \frac{-10296646794751058791702291645525069758579176693276074359410274125 59}{1213431795433441671982245442320992150323458599190996413180912100 0}.$$

Results and timings for other equations are given in tables 7.2 and 7.3.

Prime	r	s	t
317	95450823979	7160774570	9156486995910318703809
461	475038539	240767770	1264941190283659521
509	-46435	-22814	11938856913
1213	-1967568	-279136	4727718793472
1637	22342650786439	31284432683530	396018444437651200481399455 29
2917	-3606612172600350805831	2297938793190572014410	285493596261203315374167041270080026045845689
3389	-23029235	544714	530626881686193
4861	-624048447324687171	105058598975710510	8624606940847543897268692634895882 41
5701	-891892998249	7982043770	795487666473029298260249
6229	805396221085862288	-13345974078067048800	1405755268598939081029928261671730059 4944
6829	-3662456329139	-646466137590	37049278877115755537354929
6869	-25693014985	13184087914	1442123369049233982 4473
7333	2540103	142898	6684873520697
7349	2467754256268265	1242361720206386	1324554226101266337126445838689 53
7901	-6692559322050373286771	28096166595209575330	447904052402889344684580941542436511412698 09
8221	5040901127439656154292291	-351002834393950286534010	2775768790160691297242577362110455509949647923 3681
8269	212753669	17963070	53942428205174689
8293	243070279807	-167562807906	2557566928461219068243977
8501	-69648329492929110209	-5090651727025979150	5407418802446792826964919179755610882569
8941	70324600380826807669	-9742286949384739230	1024703971144529135725405645583919881856 1
12517	277117108464	36593262880	168349446566149444200704
16421	610855627205646192009189963079	-191070068137213009928944680 70	3760658317874183124724067087447337433456804418494836339660 09
17293	-22117	-858	498649057
17509	-25683719	1433790	713538950415289
17789	-8283605164969627171	251461725521118550	691344588204325938209296804778436930 09

Table 7.2: Values of r, s and t for the generators of various elliptic curves of the form $y^2 = x^3 + px$

Prime	Height	N	p6sim		p4quart		p4sim		descent
			P	time	P	time	P	time	time
317	50.6111	24	7	1.313	11	1.625	11	1.547	36.548
461	41.7531	14	5	0.250	7	0.141	7	0.187	2.891
509	23.2754	5	5	1.985	5	2.594	5	1.126	1.953
1213	23.8043	5	5	1.984	5	0.984	5	0.328	3.203
1637	65.8544	62	11	6.984	17	24.593	11	7.843	2.438
2917	102.3829	602	17	9.063	47	285.545	29	17.752	3.875
3389	33.9196	9	5	0.188	7	1.109	5	0.218	1.109
4861	82.8910	178	11	1.063	29	14.609	17	5.218	5.502
5701	55.0360	32	7	1.311	11	6.281	11	6.171	43.891
6229	86.8988	229	11	4.781	29	48.438	19	10.765	2.281
6829	59.0004	40	7	1.813	13	5.641	11	1.782	2.859
6869	51.0432	25	7	1.735	11	11.844	11	8.954	1.688
7333	29.6294	7	5	0.875	5	1.516	5	0.734	4.636
7349	73.9841	102	11	3.583	19	24.952	13	2.097	1.355
7901	100.5113	535	17	21.891	47	262.829	29	102.503	5.658
8221	113.9832	1242	23	50.374	67	1275.603	37	137.407	4.313
8269	38.6879	12	5	0.531	7	0.375	7	0.392	3.953
8293	56.2115	34	7	0.796	11	8.203	11	1.75	1.875
8501	91.6325	308	13	3.766	37	136.653	23	33.109	19.784
8941	92.2795	320	13	0.500	37	164.702	23	26.328	2.610
12517	48.0819	21	5	0.203	11	2.013	7	0.451	30.383
16421	137.2290	5308	37	95.748	137	1798.708	67	228.25	8.626
17293	20.1044	4	3	0.156	5	2.093	5	1.203	4.563
17509	34.3320	9	5	0.531	7	4.016	5	0.437	3.250
17789	87.1801	233	11	6.172	29	21.548	19	29.908	2.344

Table 7.3: Run times (in seconds) for `p6sim`, `p4quart` and `p4sim`

7.2 Analysis of results

7.2.1 The p6sim method

The expected running time of the `p6sim` method is $O(N^{2/3})$, so for solutions $\boldsymbol{w}$ with $|w_i| \leq N$, we expect the run time to be given by $T = \alpha N^{\beta}$ for some constant α and with $\beta = \frac{2}{3}$. After taking logarithms of the run time and N for 25 different runs of the algorithm, the linear correlation coefficient between the two datasets was found to be 0.793, indicating a fairly strong positive linear correlation. By the method of least squares, the best fit straight line for this data set was found to have the equation $\ln(T) = 0.685\ln(N) - 2.126$. This has been plotted together with the actual timings for the 25 runs of the algorithm in figure 7.1. The relationship between T and N is therefore given by the equation $T = 0.119N^{0.685}$.

The slope of the straight line is expected to have gradient $\beta = \frac{2}{3}$. Since the variance in the population is unknown, we estimate it by the variance in the sample.

$$s^2 = \frac{\left(\sum \ln(T_i)^2 - \frac{(\sum \ln(T_i))^2}{n}\right) - b^2\left(\sum \ln(N_i)^2 - \frac{(\sum \ln(N_i))^2}{n}\right)}{n-2}$$

$$= 1.084.$$

The confidence interval is given by

$$CI = 0.685 \pm 0.263.$$

Therefore, we are 95% confident that the true slope lies in the range (0.422, 0.949). The expected value of $\frac{2}{3}$ lies within this confidence interval.

Assuming that the running time of the `p6sim` method is given by the equation $T = 0.119N^{2/3}$, it would be possible to search for solutions to the simultaneous equations for a curve with a generator of expected height 247 in one hour, or 323 in one day.

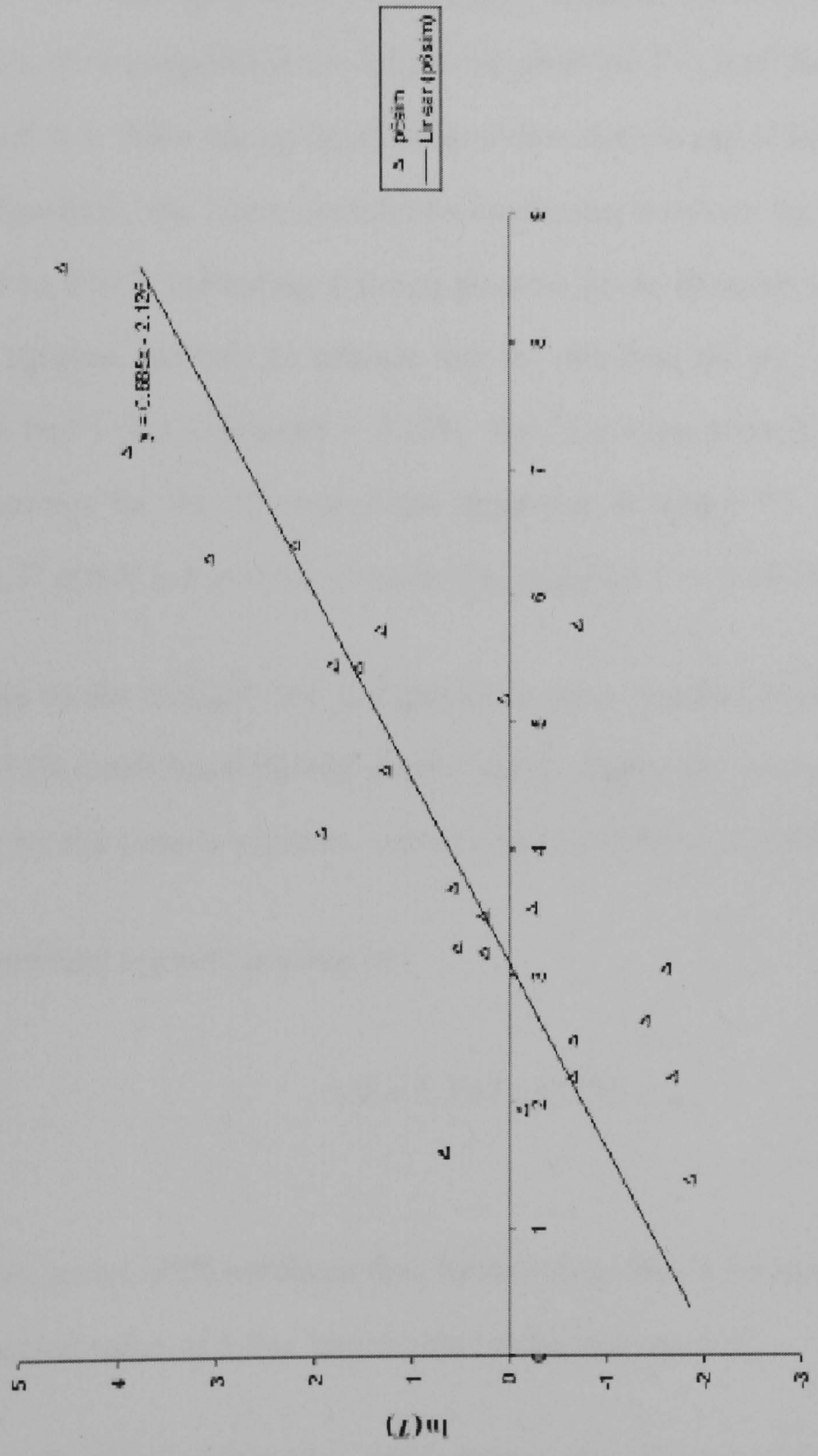

Figure 7.1: Run times for the `p6sim` method

7.2.2 The `p4quart` method

The expected running time of the `p4quart` method is $O(N)$, so for solutions $\boldsymbol{w}$ with $|w_i| \leq N$, we expect the run time to be given by $T = \alpha N^{\beta}$ for some constant α and with $\beta = 1$. After taking logarithms of the run time and N for 25 different runs of the algorithm, the linear correlation coefficient between the two datasets was found to be 0.917, indicating a strong positive linear correlation. By the method of least squares, the best fit straight line for this data set was found to have the equation $\ln(T) = 1.169 \ln(N) - 2.239$. This has been plotted together with the actual timings for the 25 runs of the algorithm in figure 7.2. The relationship between T and N is therefore given by the equation $T = 0.107N^{1.169}$.

The slope of the straight line is expected to have gradient $\beta = 1$. We can construct a 95% confidence interval for the slope. Again, we estimate the population variance by the sample variance, which is found to be $s^2 = 1.001$.

The confidence interval is given by

$$CI = 1.169 \pm 0.172.$$

Therefore, we are 95% confident that the true slope lies in the range (0.997, 1.340). The expected value of 1 lies within this confidence interval.

Assuming that the running time of the `p4quart` method is given by $T = 0.107N$, it would be possible to search for solutions to the simultaneous equations for a curve with a generator of expected height 166 in one hour, or 217 in one day.

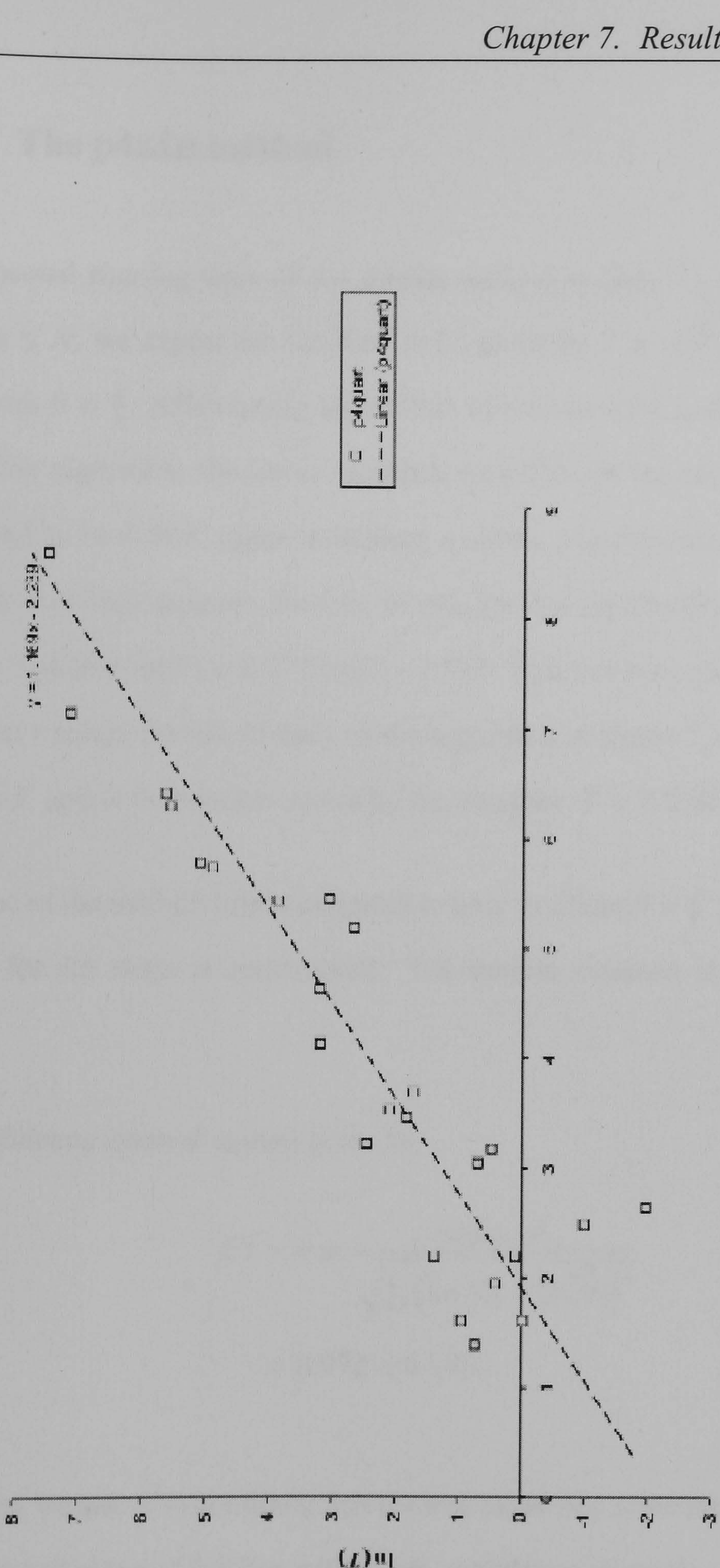

Figure 7.2: Run times for the p4quart method

7.2.3 The `p4sim` method

The expected running time of the `p4sim` method is $O(N^{4/5})$, so for solutions $\boldsymbol{w}$ with $|w_i| \leq N$, we expect the run time to be given by $T = \alpha N^{\beta}$ for some constant α and with $\beta = \frac{4}{5}$. After taking logarithms of the run time and N for 25 different runs of the algorithm, the linear correlation coefficient between the two datasets was found to be 0.903, again indicating a strong positive linear correlation. By the method of least squares, the best fit straight line for this data set was found to have the equation $\ln(T) = 0.972 \ln(N) - 2.578$. This has been plotted together with the actual timings for the 25 runs of the algorithm in figure 7.3. The relationship between T and N is therefore given by the equation $T = 0.076N^{0.972}$.

The slope of the straight line is expected to have gradient $\beta = \frac{4}{5}$. A 95% confidence interval for the slope is constructed. The sample variance is found to be $s^2 = 0.830$.

The confidence interval is then given by

$$CI = b \pm \frac{t_{0.025,n-2}s}{\sqrt{\sum \left(\ln(N_i) - \ln(\bar{N})\right)^2}}$$

$$= 0.972 \pm 0.185.$$

Therefore, we are 95% confident that the true slope lies in the range (0.786, 1.157). The expected value of 0.8 lies within this confidence interval.

Assuming that the running time of the `p4sim` method is given by $T = 0.076N^{4/5}$,

it would be possible to search for solutions to the simultaneous equations for a curve with a generator of expected height 215 in one hour, or 278 in one day.

The graphs for `p6sim`, `p4quart` and `p4sim` are superimposed in figure 7.4.

Figure 7.3: Run times for the p4sim method

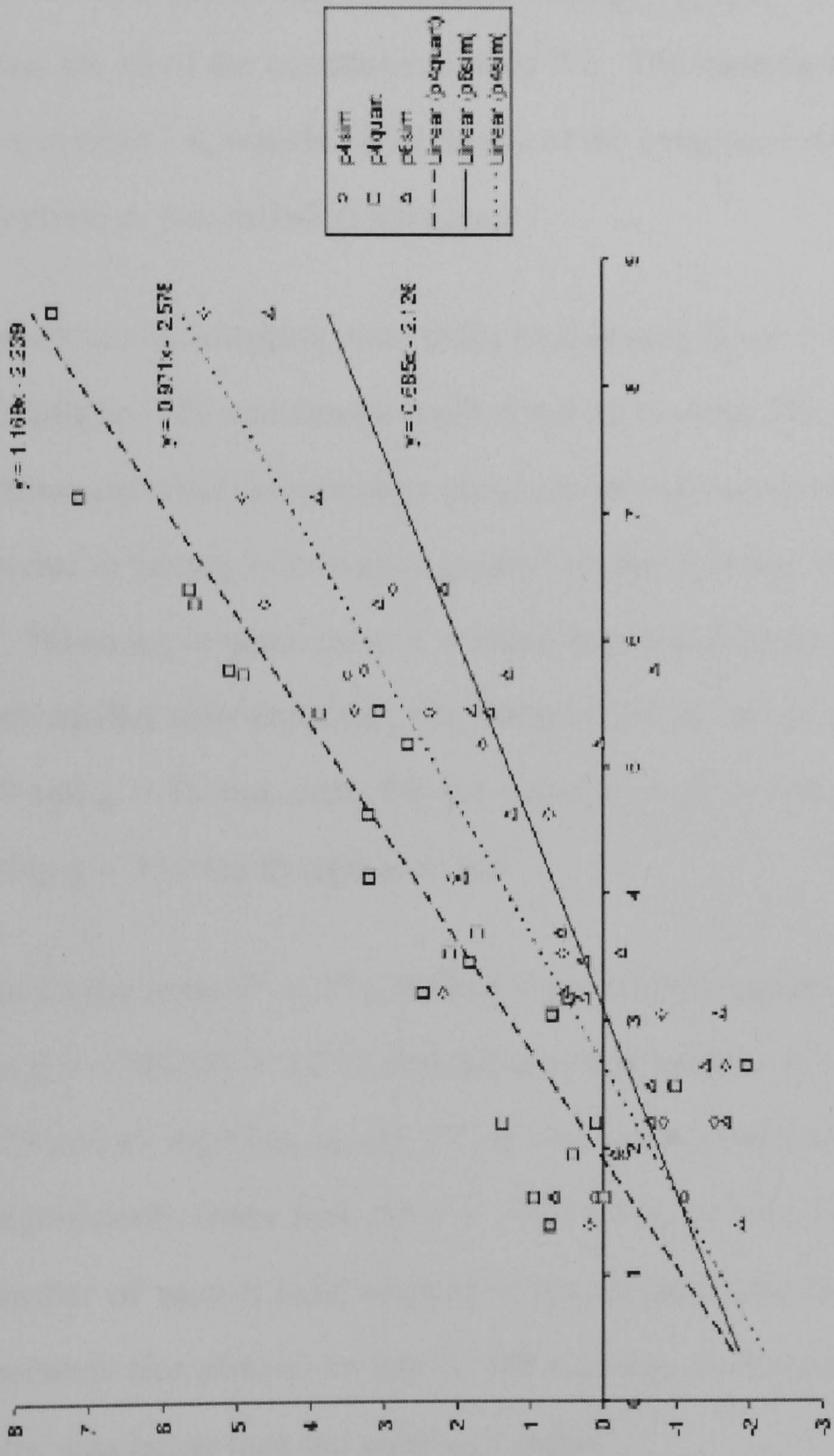

Figure 7.4: Run times for p6sim, p4quart and p4sim

7.2.4 The `composite` method

The `composite` method has been implemented as a Maple worksheet (appendix E), and run for all of the equations in table 7.2. The running times (in seconds) are shown in table 7.4, together with details of the composite modulus used. Note that each prime p_i was raised to the power 1.

These results are encouraging, with many times being faster than the corresponding run of `p6sim`. The searches for eight of the curves took longer than `p6sim`, but in six of these cases the composite modulus constructed was larger than necessary. This was due to having to avoid using small primes dividing the elements of the solution. However, in some cases a solution was found using a composite modulus much smaller than expected. For example, for the curve $Y^2 = X^3 + 6229X$, $q_0 = 119$ and $q = 38$ was used. For the curve $Y^2 = X^3 + 8501X$, a solution was found using $q = 35$ even though $q_0 = 145$.

The result for the curve $Y^2 = X^3 + 16421X$ is particularly impressive. The modulus used was $q = (29)(37) = 1073$, and the searches modulo p_i found 20 solutions modulo 29 and 40 solutions modulo 37, giving a total of 800 solutions modulo q. This is significantly fewer than the $q = 1073$ solutions expected. The reduction in the number of vectors used resulted in the running time being reduced from 95.748 seconds (for `p6sim`) to just 12.580 seconds, even though the value of q constructed was larger than the value q_0 needed.

Direct comparison of the results from `composite` with the results from `p6sim` is difficult as, in many cases, the composite number used is either much bigger or

Prime	Height	N	q_0	p_1	p_2	p_3	q	time
317	50.6111	24	27	7	11		77	1.203
461	41.7531	14	19	3	11	13	429	0.984
509	23.2754	5	10	3	5		15	0.484
1213	23.8043	5	10	3	7		21	0.891
1637	65.8544	62	50	7	11		77	3.422
2917	102.3829	602	227	13	19		247	7.657
3389	33.9196	9	14	3	5		15	0.797
4861	82.8910	178	101	11	17		187	0.593
5701	55.0360	32	32	5	7		35	0.906
6229	86.8988	229	119	2	19		38	1.533
6829	59.0004	40	38	5	7		35	0.781
6869	51.0432	25	28	5	11		55	2.390
7333	29.6294	7	12	5	11		55	1.734
7349	73.9841	102	70	7	11		77	1.172
7901	100.5113	535	210	17	19		323	11.156
8221	113.9832	1242	367	23	29		667	21.577
8269	38.6879	12	17	3	5		15	0.223
8293	56.2115	34	34	3	7	13	273	2.978
8501	91.6325	308	145	5	7		35	1.945
8941	92.2795	320	149	11	13		143	1.184
12517	48.0819	21	25	3	7		21	0.715
16421	137.2290	5308	967	29	37		1073	12.580
17293	20.1044	4	8	3	5		15	0.293
17509	34.3320	9	14	5	7		35	0.200
17789	87.1801	233	121	2	5	19	190	3.493

Table 7.4: Run times (in seconds) for the `composite` method

much smaller than theoretically necessary. However, logarithms of the timings for the runs of `composite` have been plotted against $\ln(q)$, together with logarithms of the timings from `p6sim` plotted against $\ln(P^2)$. This graph is shown in figure 7.5.

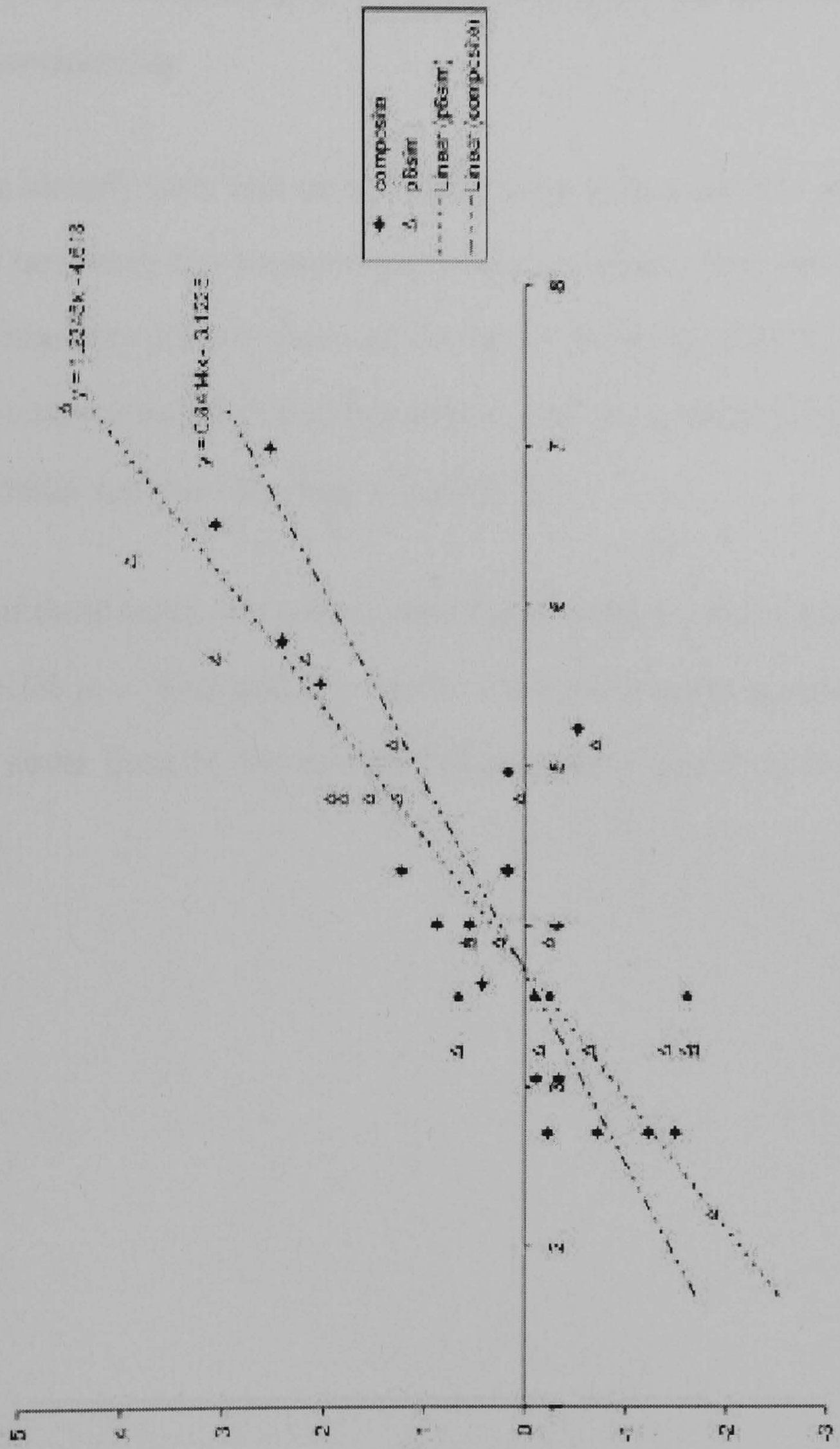

Figure 7.5: Run times for composite

7.2.5 Using a smaller auxiliary prime

Using a smaller auxiliary prime is sometimes possible, and this can speed up the search considerably.

We have already seen that using `composite` with a smaller modulus than theoretically necessary can sometimes produce a solution. Searches using `p6sim` with smaller auxiliary primes were run for the 25 curves in table 7.2. The searches for 11 of the curves failed to find a solution with any smaller prime. Results for the 14 successful searches are shown in table 7.5.

In four of these cases, the search took longer using a smaller prime. However, the searches for $p = 7901$ and $p = 16421$ were particularly successful, reducing the running times from 21.891 and 95.748 seconds respectively to 1.265 and 14.312 seconds.

Prime	Height	N	Original		Smaller	
			P	time	P	time
317	50.6111	24	7	1.313	3	0.620
461	41.7531	14	5	0.250	3	0.849
2917	102.3829	602	17	9.063	11	4.172
3389	33.9196	9	5	0.188	3	0.267
5701	55.0360	32	7	1.311	5	1.152
7333	29.6294	7	5	0.875	3	0.497
7901	100.5113	535	17	21.891	7	1.265
8293	56.2115	34	7	0.796	3	0.764
8501	91.6325	308	13	3.766	7	3.421
8941	92.2795	320	13	0.500	5	0.800
12517	48.0819	21	5	0.203	3	0.679
16421	137.2290	5308	37	95.748	31	14.312
17509	34.3320	9	5	0.531	3	0.255
17789	87.1801	233	11	6.172	5	1.116

Table 7.5: Run times (in seconds) using smaller auxiliary primes

7.2.6 Other quadric intersections

Although the timings given so far have all been for curves of the form $y^2 = x^3 + px$, where $p \equiv 5 \pmod 8$ is prime, the methods are equally applicable to quadric intersections arising elsewhere.

As part of their investigations into Mordell's equation [21], Gebel et al. compiled tables (available at [20]) listing generators of the Mordell-Weil group $E(\mathbb{Q})$ for curves of the form $E : y^2 = x^3 + k$ for all $|k| \leq 10000$, with the exception of $k = 7823$. The generator of this curve has height 77.6178, and was later found by Stoll [47].

We attempt a 4-descent on this curve. The curve has no torsion points, so we have to use general 2-descent.

Using `mwrank` [16], we compute $I = 0$ and $J = -211221$, and attempt to find 2-coverings of the curve $E_{I,J} : Y^2 = X^3 - 27IX - 27J = X^3 + 5702967$. We find four quartics for this pair (I, J). These are $(30, -12, 48, 116, -18)$, $(41, -16, -6, 112, -11)$, $(-11, -20, 408, 1784, 2072)$ and $(-18, -28, 312, 996, 838)$. The final three quartics are found to be equivalent to the first, so we take the covering curve $H : y^2 = -18x^4 + 116x^3 + 48x^2 - 12x + 30$. This is everywhere locally solvable, but no rational point is found.

We then perform a second descent, to produce the 4-covering given by the quadric

intersection

$$-22181252x_1^2 - 25045686x_1x_2 + 970984422x_1x_3 + 4436040816x_1x_4$$
$$+485492211x_2^2 + 4436040816x_2x_3 - 5908775364x_2x_4 - 2954387682x_3^2$$
$$-130297160358x_3x_4 - 185865980697x_4^2 = 0,$$
$$383480x_1^2 + 121176x_1x_2 - 18017478x_1x_3 - 74029302x_1x_4$$
$$-9008739x_2^2 - 74029302x_2x_3 + 142341300x_2x_4 + 7117650x_3^2$$
$$+2347020036x_3x_4 + 2915000865x_4^2 = 0.$$

This is more clearly written in matrix form as

$$M_1 = \begin{pmatrix} -22181252 & -12522843 & 485492211 & 2218020408 \\ -12522843 & 485492211 & 2218020408 & -2954387682 \\ 485492211 & 2218020408 & -2954387682 & -65148580179 \\ 2218020408 & -2954387682 & -65148580179 & -185865980697 \end{pmatrix},$$

$$M_2 = \begin{pmatrix} 383480 & 60588 & -9008739 & -37014651 \\ 60588 & -9008739 & -37014651 & 71170650 \\ -9008739 & -37014651 & 7117650 & 1173510018 \\ -37014651 & 71170650 & 1173510018 & 2915000865 \end{pmatrix}.$$

The coefficients in these equations are large, so searching for points on the intersection will be computationally difficult.

Stoll [47] proceeded to minimise the matrices, using a method due to Womack [53]. This is a p-adic process which replaces the rows of the matrices by linear

combinations of the rows. These substitutions can make the non-diagonal entries in the i^{th} row and column have absolute value at most half that of the i^{th} diagonal entry. We replace the rows of matrix M_1 first, repeating the process for $i = 1, 2, 3, 4$ until no further improvements are made. The same transformations are performed on M_2. We then reverse the roles of M_1 and M_2 and start again, terminating the procedure when no further improvements are made. In this way, we produce two new matrices, M_1' and M_2', such that $\det(M_1'x + M_2'y) = g(x)$.

The second stage in Stoll's approach was to reduce the matrices M_1' and M_2'. We choose a generator T of $\mathrm{SL}_4(\mathbb{Z})$ and compute $T^T M_1' T$ and $T^T M_2' T$. If one such T makes the matrices smaller in some sense, we apply the matrix T and repeat with another generator matrix T.

After minimising and reducing the matrices M_1 and M_2, we obtain the new matrices

$$M_1' = \begin{pmatrix} 1 & 2 & -1 & -1 \\ 2 & -2 & 0 & 0 \\ -1 & 0 & -3 & 2 \\ -1 & 0 & 2 & 1 \end{pmatrix},$$

$$M_2' = \begin{pmatrix} 1 & 0 & 0 & -3 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 3 & 1 \\ -3 & 0 & 1 & 1 \end{pmatrix}.$$

So we need to find a solution to the quadric intersection given by

$$x_1^2 + 4x_1x_2 - 2x_1x_3 - 2x_1x_4 - 2x_2^2 - 3x_3^2 + 4x_3x_4 + x_4^2 = 0,$$
$$x_1^2 - 6x_1x_4 + 2x_2^2 + 4x_2x_3 + 3x_3^2 + 2x_3x_4 + x_4^2 = 0.$$

Using the modulus $P = 11$ in the `p6sim` method, we find a solution $[1, 6, 4, 1] \pmod{P}$. This is raised to the solutions $[1, 94, 70, 56] \pmod{P^2}$, $[1, 14493, 2248, 177] \pmod{P^4}$ and $[1, 37597940, 176411657, 14818] \pmod{P^8}$. This last vector is reduced to give the solution $[1, 395159, 1027118, 14818] \pmod{P^6}$. The solution $\boldsymbol{w}$ then lies on a lattice with basis

$$\langle [1, 395159, 1027118, 14818], P^2[0, 9573, 10543, 1],$$
$$P^4[0, 9, 1, 0], P^6[0, 1, 0, 0] \rangle.$$

After LLL-reduction, we obtain the basis

$$\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4} \rangle = \langle [36, 601, -142, -525], [681, -116, -125, 142],$$
$$[56, 1513, 1137, 1200], [791, -303, 2009, -1200] \rangle.$$

Ignoring the fourth basis vector, we can write $\boldsymbol{w} = r_1\boldsymbol{b_1} + r_2\boldsymbol{b_2} + r_3\boldsymbol{b_3}$. We find that $r_1 = r_3 = 0, r_2 = 1$ gives the solution $\boldsymbol{w} = [681, -116, -125, 142]$. This takes 4.504 seconds.

This maps to the point

$$(x, y) = \left(\frac{53463613}{32109353}, \frac{2396334682019122}{1031010550078609} \right)$$

on H, and from there to the point on E with coordinates

$$x = \frac{22635821433214215021002092335177 77}{14356049770619098948547515190472 1},$$

$$y = \frac{18639815258462330562483755148559677002814477665575 6}{17200949981063533558210085259387279501597770434 81}.$$

7.3 Results for other implementations

7.3.1 Naïve searches

The three methods described in section 6.1 have been implemented as Maple worksheets. Each method was run for the elliptic curves listed in table 7.2, and the timings (in seconds) are shown in table 7.6. Note that this table gives the time at which the first solution in the desired range was found.

These times are shown in figure 7.6, along with the data for the `p6sim` method.

Using the method of least squares, the relationships between these times and the search bound N are found to be $T = 0.0000720N^{4.015}$ for the exhaustive search, $T = 0.00201N^{2.794}$ for an exhaustive search after eliminating one variable, and $T = 0.00193N^{2.084}$ for the resultant-based method. This would mean that the `p6sim` method is faster than the exhaustive search for curves with $\hat{h}(P) > 35.604$, faster than the search after eliminating one variable for curves with $\hat{h}(P) > 30.960$ and faster than the resultant-based method for curves with $\hat{h}(P) > 47.137$.

Prime	N	Exhaust	Elim	Resultant
317	24	12.703	0.297	0.567
461	14	0.157	0.328	0.043
509	5	0.094	0.094	0.114
1213	5	0.204	0.063	0.198
1637	62	1896.864	19.062	20.455
2917	602	6939530.213	8610.613	624.129
3389	9	1.047	0.109	0.341
4861	178	132126.130	467.906	148.687
5701	32	2.188	0.078	0.085
6229	229	752127.168	2547.855	533.698
6829	40	126.147	1.672	2.131
6869	25	18.311	0.468	0.880
7333	7	0.047	0.078	0.057
7349	102	11016.686	64.954	52.954
7901	535	1276794.380	1903.934	161.444
8221	1242	325221038.474	232576.050	8288.330
8269	12	5.562	0.297	0.909
8293	34	16.672	0.313	0.369
8501	308	1021849.252	2714.709	432.742
8941	320	1936876.811	4823.342	644.872
12517	21	55.280	1.376	2.955
16421	5308	70684331174.896	14420563.157	211051.350
17293	4	0.047	0.078	0.143
17509	9	1.626	0.125	0.540
17789	233	735736.245	2782.486	546.265

Table 7.6: Run times (in seconds) for the naïve searches

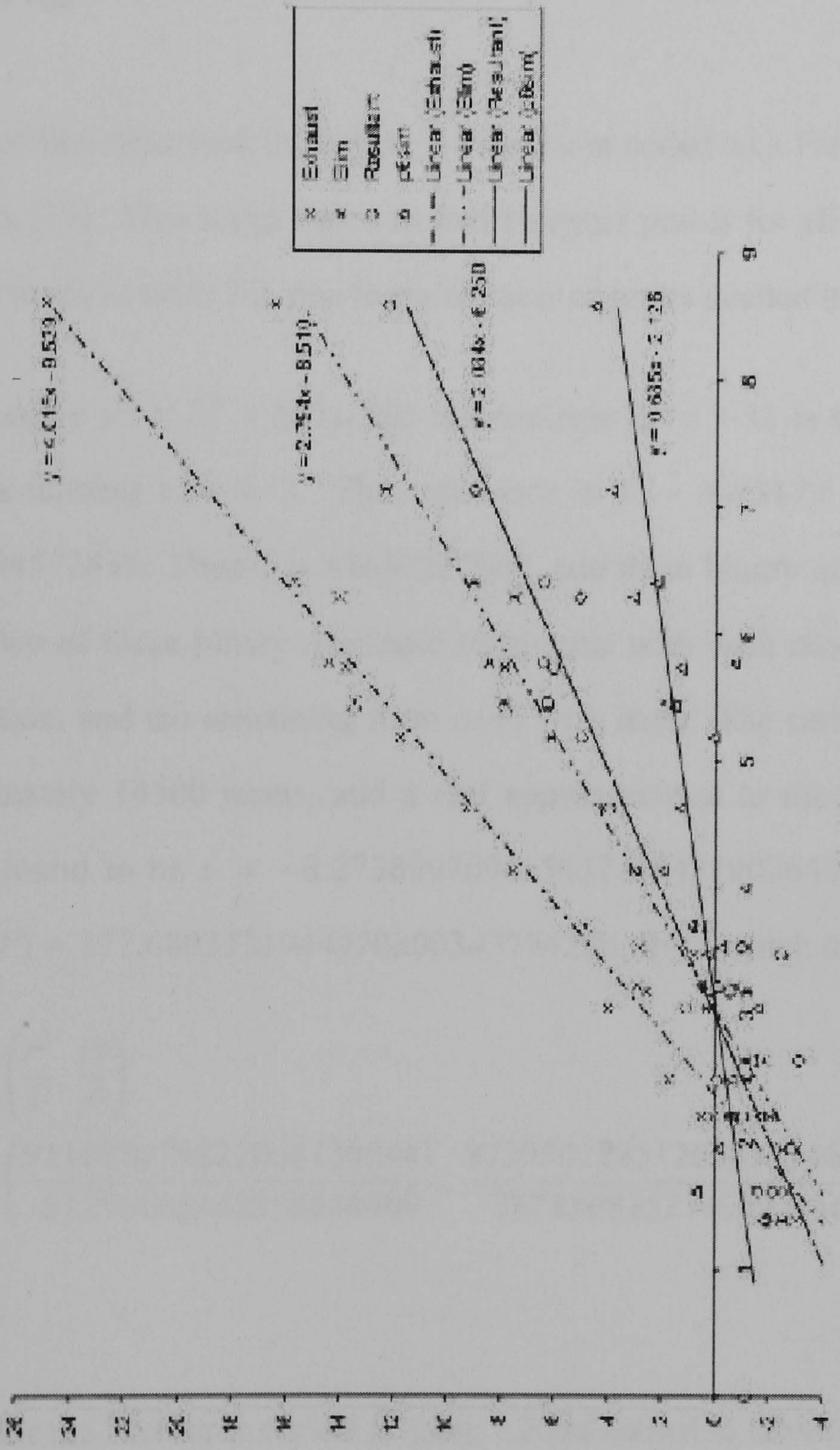

Figure 7.6: Run times for the naïve searches

7.3.2 Heegner point constructions

heegner.gp

The algorithm described in section 3.4 has been coded as a Pari/GP [2] script by Cremona [15]. This script failed to find Heegner points for all but the first three elliptic curves in table 7.2, due to the number of terms needed in the series (3.32).

For the curve $y^2 = x^3 + 317x$, the discriminant $D = -31$ is chosen, for which the class number is $h = 3$. The conductor is $N = 64(317)^2 = 6431296$, and $s = -634572431$. Then $c = 15653227988$, and three binary quadratic forms are used. Two of these binary quadratic forms pair with each other under complex conjugation, and the remaining form pairs with itself. The series (3.32) requires approximately 14500 terms, and a real approximation to the Heegner point in $\mathbb{C}/\Lambda$ is found to be $z = -8.273699709415437211431907644$. The map $\eta_E(z)$ gives $x(P) = 177.68033319449708003437942 80$, from which the rational point

$$(x,y) = \left(\frac{r^2}{s^2}, \frac{rt}{s^3}\right)$$
$$= \left(\frac{911085979827004139244 1}{5127669244235868490 0}, \frac{8739942285126383234670642958360 11}{3671808352749532616505629930 00}\right)$$

is found.

Results for the first three curves in table 7.2 are shown in table 7.7.

In all three cases, the running time was longer than the 2-descent and `p6sim` method as given in table 7.3.

Prime	r	s	t	time (secs)
317	95450823979	7160774570	91564869959103187038 09	1235.844
461	475038539	240767770	12649411902836595 21	2730.453
509	46435	22814	11938856913	3312.609

Table 7.7: Points found by a Heegner point construction

7.3.3 2-descent

`mwrank`

The C++ program `mwrank` [16] uses 2-descent to determine the rank of an elliptic curve $E/\mathbb{Q}$, find a set of points which generate $E(\mathbb{Q})/2E(\mathbb{Q})$, and finally find generators for $E(\mathbb{Q})$. If the curve has a rational 2-torsion point, the method of 2-descent via 2-isogeny is used. If there is no 2-torsion, the general 2-descent method is used.

The program was run for the curves listed in table 7.2. The bound used in the search for points on the homogeneous quartics (3.17) or (3.18) can be changed using the `-b` flag. The default value is 10, but it was necessary to change the value for some of the curves. The maximum value is 15.

For the elliptic curve $E : y^2 = x^3 + 2917x$, the isogenous curve is $E' : y^2 = x^3 - 11668x$. For $d = 2917$, we use $d_1 = 2917$. The quartic $H(2917, 0) : v^2 = 2917u^4 + 1$ has points everywhere locally, and a global solution is found.

For $d' = -11668$, we use $d_1 = -2917$ and $d_1 = -1$. The quartics $H'(-2917, 0) : v^2 = -2917u^4 + 16$ and $H'(-1, 0) : v^2 = -u^4 + 11668$ both have points everywhere

locally, but the program fails to find global points. A second descent is then carried out.

For $d_1 = -1$ and $d_3 = 1$, the quartic $(2, -424, 33720, -1191424, 15792208)$ is searched for solutions. The quartic is everywhere locally soluble, but no rational point is found within the search range. The second descent is therefore inconclusive for $d_1 = 1$.

For $d_1 = 2917$ and $d_3 = 1$, the quartic $(-27, -106, 6, 220, 108)$ is searched for solutions. Again, the quartic is everywhere locally soluble. In this instance, the search for a global solution is successful, and the point

$$(x : y : z) = (-12181 : 76394752761 : 92594)$$

is found on D. This maps to the point

$$(x : y : z) = (15039899405 : 3606612172600350805831 : 76394752761)$$

on H and from there to the point

$$(x : y : z) = (504068805486799710999198293063939 25 :$$
$$1582270768119664249710718413149289 35 :$$
$$4458518665380773296744775916870 81)$$

on E'. Finally, we recover the point

$$(x : y : z) = (298907866765975395862198444220820041647089971224480257373760840 10 :$$
$$10296646794751058791702291645525069758579176693260743594102741255 9 :$$
$$12134317995433441671982245442320992150323458599190996413180912100 0)$$

on E.

Timings and results for the curves listed in table 7.2 are given in table 7.8.

Prime	-b	r	s	t	time (secs)
317	10	95450823979	7160774570	-9156486995910318703809	1.39
461	10	475038539	240767770	-1264941190283659521	1.43
509	10	46435	22814	-11938856913	0.49
1213	10	122973	17446	-18467651537	0.49
1637	10	22342650786439	31284432683530	-396018444437651200481399455 29	1.54
2917	12	3606612172600350805831	2297938793190572014410	285493596261203315374167041270080026045845689	46.11
3389	10	23029235	544714	-530626881686193	1.52
4861	10	624048447324687171	105058598975710510	-8624606940847543897268692634895882 41	1.55
5701	10	891892998249	7982043770	-795487666473029298260249	1.55
6229	10	50337263817866393	834123379879190550	549123151796460578527315727215519554 49	1.79
6829	10	3662456329139	646466137590	-3704927887711575555373549 29	1.59
6869	10	25693014985	13184087914	-14421233690492339824473	1.56
7333	10	2540103	142898	-6684873520697	1.58
7349	10	2467754256268265	1242361720206386	-1324554226101266337126445838689 53	1.59
7901	11	6692559322050373286771	28096166595209575330	-44790405240288934468458094154243651141269809	2.11
8221	13	5040901127439656154292291	351002834393950286534010	-27757687901606912972425773621104555099496479233681	77.48
8269	10	212753669	17963070	-53942428205174689	1.56
8293	10	243070279807	167562807906	-255756692846121906824397 7	1.57
8501	10	69648329492929110209	5090651727025979150	-54074188024467928269649191797556108825 69	1.72
8941	10	70324600380826807669	9742286949384739230	-10247039711445291357254056455839198818561	1.79
12517	10	17319819279	2287078930	-657615025649021266409	1.51
16421	15	610855627205646192009189963079	191070068137213009928944680 70	-37606583178741831247240670874473374334568044184948363396600 9	731.79
17293	10	22117	858	-498649057	0.56
17509	10	25683719	1433790	-713538950415289	1.48
17789	10	8283605164969627171	251461725521118550	-691344588204325938209296804778436930 09	3.28

Table 7.8: Results from `mwrank`

Apecs

Since `p6sim` has been implemented in Maple, it makes sense to compare the timings with a Maple implementation of 2-descent. `Apecs` is a package of Maple routines written by Ian Connell [11] for computing with plane elliptic curves. It contains implementations of various algorithms associated to elliptic curves defined over $\mathbb{Z}/p\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{C}$ amongst others.

To calculate the rank unconditionally, the `RkNC(args);` command is used, where `args` is the search limit to be used while searching for rational points on the homogeneous spaces (3.17) or (3.18). By default, the value of `args` is 30. The generator(s) found after the rank has been calculated may be accessed using `RR;`.

The elliptic curve $E : y^2 = x^3 + 317x$ has the 2-isogenous curve $E' : y^2 = x^3 - 1268x$. For E', the homogeneous spaces $d_1u^4 + d_2v^4 = w^2$ are searched for solutions, where $d_1d_2 = -1268$ and d_1 is chosen from the set $\{-1, 317\}$. A second descent is performed on $-T^4 + 1268 = Y^2$ (so that $d_1 = -1$), and finds two associated quartics, both of which are locally soluble. The point $\left(\frac{263}{64}, \frac{48869}{4096}\right)$ is found on the quartic $-7T^4 + 22T^3 + 42T^2 - 22T - 7 = Y^2$. This maps to the point $(x, y) = \left(\frac{911085979827004139244 1}{5127669244235868490 0}, \frac{873994228512638323467064295836011}{367180835274953261650562993000}\right)$ on E.

`Apecs` has been run for all of the curves in table 7.2 with generators of height less than 70. If the search is unsuccessful, the value of `args` is increased to 50, and then in increments of 50 until the rank is found. The results are given in table 7.9. The timings are given only for the final (successful) value of `args`, even though many unsuccessful searches may have been carried out.

These times are shown in figure 7.7, along with the data for the `p6sim` method (including the descent).

Using the method of least squares, the relationship between these times and the expected height $\hat{h}(P)$ is found to be $T = e^{0.165\hat{h}(P)-5.269}$, while the descent and `p6sim` method has the relationship $T = e^{0.024\hat{h}(P)+0.878}$. This would mean that the `p6sim` method is faster than the apecs method for curves with $\hat{h}(P) > 43.595$.

Prime	height	`args`	time (secs)
317	50.6111	300	18.655
461	41.7531	100	0.390
509	23.2754	30	0.485
1213	23.8043	30	0.330
1637	65.8544	1350	1864.123
3389	33.9196	50	1.532
5701	55.0360	450	76.219
6829	59.0004	400	34.656
6869	51.0432	150	16.093
7333	29.6294	30	0.736
8269	38.6879	30	0.890
8293	56.2115	500	85.532
12517	48.0819	150	20.435
17293	20.1044	30	0.563
17509	34.3320	50	0.921

Table 7.9: Run times (in seconds) for `Apecs`

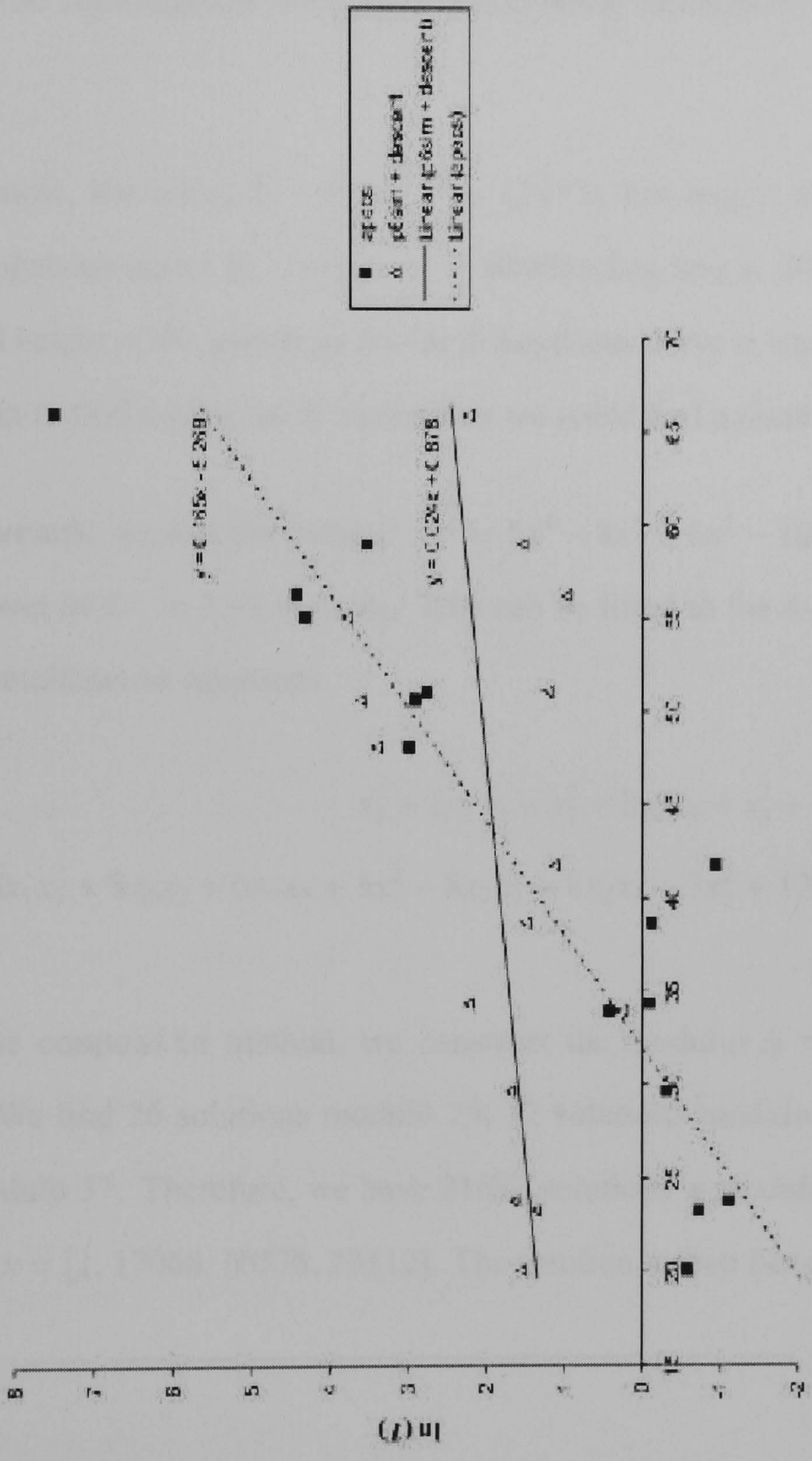

Figure 7.7: Run times for apecs

7.4 Combinations of new and existing methods

We can use combinations of the new and existing methods to deal with difficult cases.

For example, the curve $E : y^2 = x^3 + 17477x$ has height 406.479655, while the 2-isogenous curve $E' : y^2 = x^3 - 69908x$ has height 203.240. Since the expected height of the generator for the 2-isogenous curve is significantly smaller, we expect to find a point on E' faster than we could find a point on E.

Using `mwrank`, we find the curve $C : y^2 = 5x^4 - 8x^3 + 6x^2 - 108x + 3538$, which is a 2-cover of E', in 2.99 seconds. This can be lifted to the 4-cover of E' given by the simultaneous equations

$$x_1^2 + 2x_1x_3 - x_2^2 - 2x_2x_4 - x_3^2 + 2x_3x_4 + x_4^2 = 0,$$
$$x_1^2 + 8x_1x_2 + 8x_1x_3 + 6x_1x_4 + 3x_2^2 - 8x_2x_3 + 8x_2x_4 + 3x_3^2 + 12x_3x_4 - 9x_4^2 = 0.$$

Using the `composite` method, we construct the modulus $q = (29)(31)(37) = 33263$. We find 26 solutions modulo 29, 32 solutions modulo 31, and 26 solutions modulo 37. Therefore, we have 21632 solutions $\boldsymbol{x}$ modulo q, including the solution $\boldsymbol{x} = [1, 17068, 10578, 20112]$. The solution $\boldsymbol{w}$ then lies on the lattice with

basis

$$\langle [1, 1946434776, 236067792843 39, 329947646601 24],$$

$$q[0, 402349095, 925194551, 1],$$

$$q^2[0, 3924, 1, 0],$$

$$q^3[0, 1, 0, 0]\rangle.$$

Using Maple's `lattice` command, we obtain the basis

$$\begin{aligned}\langle \boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}, \boldsymbol{b_4}\rangle = \langle &[-2436965, -1331634, -222345, 3285145],\\ &[-1619713, 2678012, 4438746, 1216243],\\ &[4771457, -304500, -1003632, 3646692],\\ &[1356616, -6205223, 7577815, -1544223]\rangle.\end{aligned}$$

However, the basis returned from the `lattice` command in this instance fails to satisfy the properties of a LLL-reduced basis (see theorem 5.2.2). In particular, it is easy to verify that $\prod_{i=1}^{4} \|\boldsymbol{b}_i\| > 2^3\Delta(\mathcal{L})$. We cannot assume that $r_4 = 0$, so we have $\boldsymbol{w} = r_1\boldsymbol{b_1} + r_2\boldsymbol{b_2} + r_3\boldsymbol{b_3} + r_4\boldsymbol{b_4}$. A solution is found with $r_1 = 43$, $r_2 = 206$, $r_3 = -170$, and $r_4 = -189$. This gives the solution $\boldsymbol{w} = [-1505998487, 1718962357, -356768754, 63727800]$.

Determining the lattice basis and reducing the lattice took 52047.181 seconds (approximately 14.5 hours), although the coefficients in the linear combination were found by other means.

This solution maps to the point

$$(x,y) = \left(\frac{2219550473166045936937}{250868991864889474945}, \frac{\text{-}211179099389193092827807752985561160916223 3}{125870502158611963119955329721275565506 05}\right)$$

on C, and from there to the point $(x,y) = \left(\frac{R}{S^2}, \frac{T}{S^3}\right)$ on E', where

R =194853747063276011997739500260507110644230716190772669814688
825673077046648107221121232 1325,

S =65794607206186944852826285634714515368428551,

T =220125441873403357833217386324407559868669946844647347114789
521716963237877770631271335007340918457008213835057013464965
112339093239325 5.

We then use the 2-isogeny to recover the point $(x,y) = \left(\frac{r^2}{s^2}, \frac{rt}{s^3}\right)$ on E, where

r =119284061169164623075564480842594996123764020167292972929929
984661440898801116929992771 11,

s =138944458944682729095492071510818262932987342600225784522930
703622543499294414828114383 0,

t =292203975419724482428599539551835041801177915980818301067406
885580364308301180799437937007688519262155931753460726161052
471043227400627987499297766938094594613089015507897139929.

7.5 Limitations and further work

The current implementations are for pairs of simultaneous homogeneous quadratic equations in four variables, or for a single homogeneous quartic equation in three variables, but could be adapted for other forms of equations.

We have already seen that there is no benefit in taking a complete set of solutions modulo a higher power of P (see section 6.2), and that we cannot take more than 3 terms in the Taylor series expansions for equations of degree 2.

For a pair of homogeneous equations of degree d, the Taylor series expansion of $Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t})$ is given by

$$Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t}) = \sum_{n=1}^{d+1} \lambda^{d-n+1} p^{2(n-1)} f_{[i,n-1]}(\boldsymbol{t}),$$

where $f_{[i,k]}$ is a polynomial of degree k depending on Q_i.

Writing $\boldsymbol{t}$ as $\lambda \boldsymbol{u} + P^2 \boldsymbol{m}$, we have

$$Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t}) = \sum_{n=1}^{d+1} \left(\lambda^{d-n+1} P^{2(n-1)} \sum_{j=1}^{n} \left(P^{2(j-1)} \gamma^{n-j} f_{[i,j-1,n]}(\boldsymbol{m}) \right) \right),$$

where $f_{[i,k,n]}$ is a homogeneous polynomial of degree k depending on n and Q_i.

For a quadric intersection, we have $d = 2$, so we may only use the first three terms of the Taylor series expansion as the fourth and subsequent terms will be zero.

Therefore, we have

$$Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t}) = \lambda^2 f_{[i,0,1]}(\boldsymbol{m}) + P^2 \lambda\gamma f_{[i,0,2]}(\boldsymbol{m}) + P^4 \lambda f_{[i,1,2]}(\boldsymbol{m}) + P^4\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) + P^6\gamma f_{[i,1,3]}(\boldsymbol{m}) + P^8 f_{[i,2,3]}(\boldsymbol{m}) = 0.$$

The `p6sim` method considers these equations modulo P^6, so that

$$\lambda^2 f_{[i,0,1]}(\boldsymbol{m}) + P^2 \lambda\gamma f_{[i,0,2]}(\boldsymbol{m}) + P^4 \lambda f_{[i,1,2]}(\boldsymbol{m}) + P^4\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) \equiv 0 \pmod{P^6}.$$

The first term is constructed to be zero modulo P^6 by the choice of arbitrary modulus for the solution $\boldsymbol{y}'$, and vectors are adjusted to make the second term divisible by P^6. Therefore we solve the simultaneous congruences

$$P^4 \lambda f_{[i,1,2]}(\boldsymbol{m}) + P^4\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) \equiv 0 \pmod{P^6}$$

$$\Rightarrow \lambda f_{[i,1,2]}(\boldsymbol{m}) + \gamma^2 f_{[i,0,3]}(\boldsymbol{m}) \equiv 0 \pmod{P^2}.$$

Since we have two equations and the coefficients of γ^2 are constant, we can eliminate γ. We then use the fact that $P \nmid \lambda$ to leave a linear equation in $\boldsymbol{m}$ which can easily be solved.

For equations of degree $d > 2$, we might be tempted to use the first $d + 1$ terms of the Taylor series expansion in order to find $\boldsymbol{m}$. This would mean that our lattice had a larger determinant, so that we could take a smaller prime P for the search bound N. However, this is not practical.

For example, when $d = 3$, having found an arbitrary solution modulo P^8, we would need to solve the simultaneous equations

$$\begin{aligned} Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t}) = & \lambda^3 f_{[i,0,1]}(\boldsymbol{m}) + P^2\lambda^2\gamma f_{[i,0,2]}(\boldsymbol{m}) + P^4\lambda^2 f_{[i,1,2]}(\boldsymbol{m}) \\ & + P^4\lambda\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) + P^6\lambda\gamma f_{[i,1,3]}(\boldsymbol{m}) + P^8\lambda f_{[i,2,3]}(\boldsymbol{m}) \\ & + P^6\gamma^3 f_{[i,0,4]}(\boldsymbol{m}) + P^8\gamma^2 f_{[i,1,4]}(\boldsymbol{m}) + P^{10}\gamma f_{[i,2,4]}(\boldsymbol{m}) \\ & + P^{12} f_{[i,3,4]}(\boldsymbol{m}) = 0. \end{aligned}$$

Congruence modulo P^8 gives

$$\begin{aligned} Q_i(\lambda \boldsymbol{y}' + P^2 \boldsymbol{t}) = & \lambda^3 f_{[i,0,1]}(\boldsymbol{m}) + P^2\lambda^2\gamma f_{[i,0,2]}(\boldsymbol{m}) + P^4\lambda^2 f_{[i,1,2]}(\boldsymbol{m}) \\ & + P^4\lambda\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) + P^6\lambda\gamma f_{[i,1,3]}(\boldsymbol{m}) + P^6\gamma^3 f_{[i,0,4]}(\boldsymbol{m}) \equiv 0 \pmod{P^8}. \end{aligned}$$

The first term is divisible by P^8 and vectors would be adjusted to make the second term 0 modulo P^8. Then we would be required to solve

$$\begin{aligned} & P^4\lambda^2 f_{[i,1,2]}(\boldsymbol{m}) + P^4\lambda\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) + P^6\lambda\gamma f_{[i,1,3]}(\boldsymbol{m}) + P^6\gamma^3 f_{[i,0,4]}(\boldsymbol{m}) \equiv 0 \pmod{P^8} \\ \Rightarrow\ & \lambda^2 f_{[i,1,2]}(\boldsymbol{m}) + \lambda\gamma^2 f_{[i,0,3]}(\boldsymbol{m}) + P^2\lambda\gamma f_{[i,1,3]}(\boldsymbol{m}) + P^2\gamma^3 f_{[i,0,4]}(\boldsymbol{m}) \equiv 0 \pmod{P^4} \end{aligned}$$

It is not clear how these simultaneous congruence equations could be solved for $\boldsymbol{m}$, as there is no simple method by which we could eliminate γ.

Therefore, for a pair of simultaneous equations in any degree $d \geq 2$, we can only take three terms in the Taylor series expansion, so that `p6sim` is the optimal method.

We can also adapt the methods for equations in m variables. This will clearly affect the size and determinant of the lattice basis, the size of prime needed, and the running time.

We eliminate one variable to leave an equation in $m - 1$ variables, and then set the first component to 1. We now have one equation in $m - 2$ variables. We loop around all possible values modulo P for $m-3$ of these, and solve for the remaining variable. So a complete set of solutions (x) modulo P with $x_1 = 1$ will be found in time $O(P^{m-3})$.

To lift each solution $\boldsymbol{x}$ to a complete set of solutions $\boldsymbol{y}$ modulo P^2, we need to solve a pair of simultaneous equations in $m - 1$ variables. We therefore obtain a solution in $m - 3$ of the variables, and run through all possibilities for these in time $O(P^{m-3})$. Thus we determine a complete set of solutions modulo P^2 in time $O(P^{2(m-3)})$.

We now consider the determinant of the resulting lattice for each method.

For `p6sim`, we have

$$\Delta(L) = \begin{vmatrix} 1 & y'_2 & y'_3 & \dots & y'_{m-1} & y'_m \\ 0 & P^2u_2 & p^2u_3 & \dots & P^2u_{m-1} & P^2 \\ 0 & P^4 & P^4c_1 & \dots & P^4c_k & 0 \\ 0 & 0 & P^6 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & P^6 & 0 \end{vmatrix} = P^{6(m-2)}.$$

To ensure that $|r_m| < 1$, we choose

$$P > \left(N^2 2^{m-1} m\right)^{\frac{m}{12(m-2)}}.$$

The running time will then be

$$\begin{aligned} O(P^{2(m-3)}) &= O\left(\left(\left(N^2 2^{m-1} m\right)^{\frac{m}{12(m-2)}}\right)^{2(m-3)}\right) \\ &= O\left(\left(N^2 2^{m-1} m\right)^{\frac{m(m-3)}{6(m-2)}}\right). \end{aligned}$$

These values will be the same for `composite`, with q approximately the size of P^2.

For `p4quart`, the determinant of the lattice is given by

$$\Delta(L) = \begin{vmatrix} 1 & y'_2 & y'_3 & \cdots & y'_{m-2} & y'_{m-1} \\ 0 & P^2 u_2 & p^2 u_3 & \cdots & P^2 u_{m-2} & P^2 \\ 0 & P^4 & 0 & \cdots & 0 & 0 \\ 0 & 0 & P^4 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & P^4 & 0 \end{vmatrix} = P^{2(2m-5)}.$$

We choose

$$P > \left(N^2 2^{m-2}(m-1)\right)^{\frac{(m-1)}{4(2m-5)}},$$

and the running time will be

$$O(P^{2(m-3)}) = O\left(\left(\left(N^2 2^{m-2}(m-1)\right)^{\frac{(m-1)}{4(2m-5)}}\right)^{2(m-3)}\right)$$
$$= O\left(\left(N^2 2^{m-2}(m-1)\right)^{\frac{(m-1)(m-3)}{2(2m-5)}}\right).$$

Finally, for `p4sim`, we have

$$\Delta(L) = \begin{vmatrix} 1 & y'_2 & y'_3 & \dots & y'_{m-1} & y'_m \\ 0 & P^2 u_2 & p^2 u_3 & \dots & P^2 u_{m-1} & P^2 \\ 0 & P^4 & 0 & \dots & 0 & 0 \\ 0 & 0 & P^4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & P^4 & 0 \end{vmatrix} = P^{2(2m-3)},$$

where

$$P > \left(N^2 2^{m-1} m\right)^{\frac{m}{4(2m-3)}}.$$

The running time will then be

$$O(P^{2(m-3)}) = O\left(\left(\left(N^2 2^{m-1} m\right)^{\frac{m}{4(2m-3)}}\right)^{2(m-3)}\right)$$
$$= O\left(\left(N^2 2^{m-1} m\right)^{\frac{m(m-3)}{2(2m-3)}}\right).$$

The `composite` method sometimes fails to find solutions for a modulus q, due to the possibility that none of the coordinates of the solution are coprime to the modulus chosen. In this instance, we would have to run the procedure again with

a different modulus. It is also not immediately obvious how to choose suitable primes in order to make the number of solutions small, other than by trial and error. When the expected height of the generator is large, so that the primes chosen are large, this is clearly time consuming. We could use smaller primes and construct a modulus with more prime factors, but this would increase the probability that each component is divisible by one of the primes chosen.

The timings given in this chapter are for the time at which a solution is found. In some cases, a solution is found late in the run, due to the solution $\boldsymbol{x}$ modulo P corresponding to the actual solution $\boldsymbol{w}$ being one of the last to be found. This could be avoided by considering $Q_i(a, b, c, d)$ for various permutations (a, b, c, d) of the vector $\boldsymbol{x} = [x_1, x_2, x_3, x_4]$ simultaneously, as in Womack's method for searching for solutions to a ternary quartic [53].

Finally, the difference between the running times for similar methods written in C++ and Maple can be seen from the timings for `mwrank` and `apecs`. It is therefore necessary to implement the methods in a faster programming language.

Appendix A

descent.mws

Determine pairs of equations which are equivalent to the equation Y^2=X(X^2+p) where p = 5 (mod 8) is prime

```
> restart:
> with(GaussInt):
> readlib(lattice):
> interface(showassumed=0):
```

Enter a value for p

```
> p:=5:

> starttime:=time():
```

Step 1. Find u and v such that p = u^2 + 4*v^2 with gcd(u,v)=1, u and v odd, and v = 1 (mod 4)

```
> n:=(p-5)/8:
> j:=(-2)^(1+2*n) mod p:
```

```
> reducedvectors:=lattice([[j,1],[p,0]],'integer'):
> if reducedvectors[1][1] mod 2 = 0 then
> v:=reducedvectors[1][1]/2:
> u:=reducedvectors[1][2]:
> else
> u:=reducedvectors[1][1]:
> v:=reducedvectors[1][2]/2:
> fi:
> if u<0 then
> u:=-u:
> fi:
> if v mod 4=3 then
> v:=-v:
> fi:
```

Step 2. The original equation is equivalent to the following

```
> Q:=b^2+v*m^2-v*l^2-u*l*m:
```

Step 3. Find an equivalent quadratic form: -LM + p*N^2 = 0

3.1 Find a small coprime solution (a1,a2,a3) to Q(b,l,m) = 0

```
> a1:=0:
> a2:=0:
> a3:=0:
> bsoln:=solve(Q,b)[1]:
> stopflag:=false:
> for x2 from 0 to 15 while stopflag=false do
> for x3 from -15 to 15 while stopflag=false do
> if (x2<>0 or x3<>0) then
> btest:=simplify(subs([l=x2,m=x3],bsoln)):
> if type(btest,integer) then
> a1:=btest:
```

```
> a2:=x2:
> a3:=x3:
> stopflag:=true:
> Common:=igcd(a1,a2,a3);
> a1:=a1/Common;
> a2:=a2/Common;
> a3:=a3/Common;
> fi:
> fi:
> od:
> od:
> if (a1=0 and a2=0 and a3=0) then
> printf(' No solution to search 1'):
> else
```

3.2 Set up the matrix M with (a1,a2,a3) as the first column, such that det(M) = 1

```
> if igcd(a2,a3)=1 then
> c1:=1;
> c2:=0;
> c3:=0;
> b1:=1:
> igcdex(a2,a3,'b3','b2'):
> b3:=-b3:
> elif igcd(a1,a2)=1 then
> c1:=0:
> c2:=0:
> c3:=1:
> b3:=1:
> igcdex(a1,a2,'b2','b1'):
> b2:=-b2:
> else
> b1:=0:
> b2:=1:
> b3:=0:
```

```
> c2:=1:
> igcdex(a1,a3,'c3','c1'):
> c3:=-c3:
> fi:
```

3.3 Substitution (x1,x2,x3) = M(X1,X2,X3). Then Q(x1,x2,x3) = H(X1,X2,X3). Determine the coefficients in H.

```
> eqb:=b=a1*X1+b1*X2+c1*X3;
> eql:=l=a2*X1+b2*X2+c2*X3;
> eqm:=m=a3*X1+b3*X2+c3*X3;
> H:=simplify(subs([eqb,eql,eqm],Q));
> r2:=coeff(coeff(H,X1),X2);
> r3:=coeff(coeff(H,X1),X3);
```

3.4 Find s2, s3 such that r2*s3-r3*s2 = 1

```
> igcdex(r2,r3,'s3','s2'):
> s2:=-s2:
```

3.5 Linear substitutions (Y1,Y2,Y3) = M2(X1,X2,X3) where M2 is matrix with rows ([1,0,0],[0,r2,r3],[0,s2,s3]). Then Q(x1,x2,x3) = H(X1,X2,X3) = Y1*Y2 + H1(Y2,Y3) (H1 is a quadratic in Y2,Y3)

```
> Y1:=X1;
> Y2:=r2*X2+r3*X3;
> Y3:=s2*X2+s3*X3;
> eq:=expand(Y1*Y2+k1*Y2^2+k2*Y2*Y3+k3*Y3^2);
```

3.6 Equate coefficients in Q(x1,x2,x3) and H(X1,X2,X3)

```
> eq1:=coeff(eq,X2,2)=coeff(H,X2,2);
> eq2:=coeff(coeff(eq,X2),X3)=coeff(coeff(H,X2),X3);
> eq3:=coeff(eq,X3,2)=coeff(H,X3,2);
> k:=solve({eq1,eq2,eq3},{k1,k2,k3});
> k1:=subs(k,k1):
> k2:=subs(k,k2):
> k3:=subs(k,k3):
```

3.7 Final substitution

```
> M:=-(Y1+k1*Y2+k2*Y3):
> L:=Y2:
> N:=Y3:
> solns:=solve({eqb,eql,eqm},{X1,X2,X3}):
> M:=subs(solns,M);
> L:=subs(solns,L);
> N:=subs(solns,N);
> if coeff(N,b)<0 then
> N:=-N:
> fi:
```

Step 4. Choose sign of b so that p does not divide M

4.1 There are two possibilities for M,L,N

```
> eqM:=Z1=M:
> eqL:=Z2=L:
> eqN:=Z3=N:
> solns:=solve({eqM,eqL,eqN},{b,l,m}):
> solnsplus:=subs([Z2=k3*theta^2,Z1=phi^2,Z3=theta*phi],solns);
> solnsminus:=subs([Z2=-k3*theta^2,Z1=-phi^2,Z3=-theta*phi],solns);
```

4.2 One of these sets of solutions will be 2-adically incompatible with b=1, l=3, m=0 mod 4 - decide which is correct by checking all possibilities mod 4

```
> bplus:=subs(solnsplus,b):
> bminus:=subs(solnsminus,b):
> lplus:=subs(solnsplus,l):
> lminus:=subs(solnsminus,l):
> mplus:=subs(solnsplus,m):
> mminus:=subs(solnsminus,m):
> plusflag:=false:
> minusflag:=false:
> for t1 from 0 to 3 do
```

```
> for p1 from 0 to 3 do
> bpluscheck:=subs([theta=t1,phi=p1],bplus) mod 4:
> bminuscheck:=subs([theta=t1,phi=p1],bminus) mod 4:
> lpluscheck:=subs([theta=t1,phi=p1],lplus) mod 4:
> lminuscheck:=subs([theta=t1,phi=p1],lminus) mod 4:
> mpluscheck:=subs([theta=t1,phi=p1],mplus) mod 4:
> mminuscheck:=subs([theta=t1,phi=p1],mminus) mod 4:
```

1. check plus

```
> if (bpluscheck=1 and lpluscheck=3 and mpluscheck=0) then
> plusflag:=true:
> twothetamod4:=2*t1 mod 4:
> twophimod4:=2*p1 mod 4:
> fi:
```

2. check minus

```
> if (bminuscheck=1 and lminuscheck=3 and mminuscheck=0) then
> minusflag:=true:
> twothetamod4:=2*t1 mod 4:
> twophimod4:=2*p1 mod 4:
> fi:
> od:
> od:
> if (plusflag=false and minusflag=false) then
> printf(‘ 2-adic check 1 failed‘):
> else
```

4.3 Simplify these solutions if possible by making a substitution for phi

```
> if plusflag=true then
> btemp:=subs(solnsplus,b):
> ltemp:=subs(solnsplus,l):
> mtemp:=subs(solnsplus,m):
> else
> btemp:=subs(solnsminus,b):
```

```
> ltemp:=subs(solnsminus,l):
> mtemp:=subs(solnsminus,m):
> fi:
> coeffphi:=coeff(ltemp,phi,2):
> coefftheta:=coeff(coeff(ltemp,theta),phi):
> if type(coefftheta/(2*coeffphi),integer)=false then
> coeffphi:=coeff(mtemp,phi,2):
> coefftheta:=coeff(coeff(mtemp,theta),phi):
> fi:
> if type(coefftheta/(2*coeffphi),integer)=false then
> coeffphi:=coeff(btemp,phi,2):
> coefftheta:=coeff(coeff(btemp,theta),phi):
> fi:
> thetamultiple:=coefftheta/(2*coeffphi):
> if type(thetamultiple,integer)=false then
> thetamultiple:=round(thetamultiple):
> fi:
```

```
> if plusflag=true then
> correctsoln:=simplify(subs(phi=psi-thetamultiple*theta,solnsplus)):
> twopsimod4:=twophimod4-thetamultiple*twothetamod4 mod 4:
> else
> correctsoln:=simplify(subs(phi=psi-thetamultiple*theta,solnsminus)):
> twopsimod4:=twophimod4+thetamultiple*twothetamod4 mod 4:
> fi:
```

Step 5. Write l+i*m = gamma^2 and determine the corresponding equation cQ in terms of gamma, psi and theta

```
> cQ:=gamma^2-l-I*m:
> cQ:=simplify(subs(correctsoln,cQ)):
```

Step 6. Find an equivalent quadratic form: -RT + ck3*S^2 = 0 (this time working in Gaussian integers)

6.1 Find a small coprime solution (ca1,ca2,ca3) to cQ(gamma,psi,theta) = 0

```
> cQtemp:=subs(gamma=gam,cQ): #have to do this as Maple doesn't
allow us to solve for gamma
> gamsoln:=traperror(solve(cQtemp,gam)[1]):
> ca1:=0:
> ca2:=0:
> ca3:=0:
> stopflag:=false:
> searchlim:=10:
> im2:=0:
> im3:=0:
> for re2 from 0 to searchlim while stopflag=false do
> for im2 from -searchlim to searchlim while stopflag=false
do
> for re3 from -searchlim to searchlim while stopflag=false
do
> for im3 from -searchlim to searchlim while stopflag=false
do
> gamtest:=subs([psi=re2+I*im2,theta=re3+I*im3],gamsoln):
> gamtest:=simplify(gamtest):
> coeff1:=traperror(coeff(gamtest,I,0)):
> coeffI:=traperror(coeff(gamtest,I)):
> if coeff1<>‘unable to compute coeff‘ then
> if coeffI<>‘unable to compute coeff‘ then
> if (type(coeff1,integer) and type(coeffI,integer)) then
> ca1:=gamtest:
> ca2:=re2+I*im2:
> ca3:=re3+I*im3:
> if (ca1<>0 or ca2<>0 or ca3<>0) then
> cd:=GIgcd(ca1,ca2,ca3):
> ca1:=ca1/cd:
> ca2:=ca2/cd:
> ca3:=ca3/cd:
> stopflag:=true:
> fi:
> fi:
> fi:
```

```
> fi:
> od:
> od:
> od:
> od:
> if (ca1=0 and ca2=0 and ca3=0) then
> printf(‘ no solution to search 2‘):
> else
```

6.2 Set up the matrix M with (ca1,ca2,ca3) as the first column such that det(M) = 1

```
> if GIgcd(ca2,ca3)=1 then
> cc1:=1;
> cc2:=0;
> cc3:=0;
> cb1:=1:
> GIgcdex(ca2,ca3,'cb3','cb2'):
> cb3:=-cb3:
> elif GIgcd(ca1,ca2)=1 then
> cc1:=0:
> cc2:=0:
> cc3:=1:
> cb3:=1:
> GIgcdex(ca1,ca2,'cb2','cb1'):
> cb2:=-cb2:
> else
> cb1:=0:
> cb2:=1:
> cb3:=0:
> cc2:=1:
> GIgcdex(ca1,ca3,'cc3','cc1'):
> cc3:=-cc3:
> fi:
```

6.3 Substitution (gamma,psi,theta) = M(cX1,cX2,cX3). Then cQ(gamma,psi,theta) = cH(cX1,cX2,cX3). Determine the coefficients in cH

```
> eqgamma:=gamma=ca1*cX1+cb1*cX2+cc1*cX3;
> eqpsi:=psi=ca2*cX1+cb2*cX2+cc2*cX3;
> eqtheta:=theta=ca3*cX1+cb3*cX2+cc3*cX3;
> cH:=simplify(subs([eqgamma,eqpsi,eqtheta],cQ));
> cr2:=coeff(coeff(cH,cX1),cX2);
> cr3:=coeff(coeff(cH,cX1),cX3);
> cCommon:=GIgcd(cr2,cr3);
> cr2:=cr2/cCommon;
> cr3:=cr3/cCommon;
```

6.4 Find cs2, cs3 such that cr2*cs3-cr3*cs2 = 1

```
> GIgcdex(cr2,cr3,'cs3','cs2'):
> cs2:=-cs2:
```

6.5 Linear substitutions (cY1,cY2,cY3) = M2(cX1,cX2,cX3) where M2 is a matrix with rows ([1,0,0],[0,cr2,cr3],[0,cs2,cs3]). Then cQ(gamma,psi,theta) = cH(cX1,cX2,cX3) = cY1*cY2 + cH1(cY2,cY3) (cH1 is a quadratic in cY2,cY3)

```
> cY1:=cX1;
> cY2:=cr2*cX2+cr3*cX3;
> cY3:=cs2*cX2+cs3*cX3;
> ceq:=expand(cY1*cY2+ck1*cY2^2+ck2*cY2*cY3+ck3*cY3^2);
```

6.6 Equate coefficients in cQ(gamma,psi,theta) and cH(cX1,cX2,cX3)

```
> eqc1:=coeff(ceq,cX2,2)=coeff(cH,cX2,2);
> eqc2:=coeff(coeff(ceq,cX2),cX3)=coeff(coeff(cH,cX2),cX3);
> eqc3:=coeff(ceq,cX3,2)=coeff(cH,cX3,2);
> ck:=solve({eqc1,eqc2,eqc3},{ck1,ck2,ck3});
> ck1:=subs(ck,ck1):
> ck2:=subs(ck,ck2):
> ck3:=subs(ck,ck3):
```

6.7 Final substitution

```
> T:=simplify(-(cCommon*cY1+ck1*cY2+ck2*cY3)):
> R:=simplify(cY2):
> S:=simplify(cY3):
> csolns:=solve({eqgamma,eqpsi,eqtheta},{cX1,cX2,cX3}):
> T:=simplify(subs(csolns,T));
> R:=simplify(subs(csolns,R));
> S:=simplify(subs(csolns,S));
```

Step 7. Choose sign of gamma so that ck3 does not divide R

7.1 There are four possible solutions

```
> eqR:=R1=R:
> eqT:=T1=T:
> eqS:=S1=S:
> eqR:=subs([gamma=gamma1,psi=psi1,theta=theta1],eqR):
> eqT:=subs([gamma=gamma1,psi=psi1,theta=theta1],eqT):
> eqS:=subs([gamma=gamma1,psi=psi1,theta=theta1],eqS):
> csolns2:=solve({eqR,eqT,eqS},{gamma1,psi1,theta1}):
> twogamma:=expand(subs(csolns2,2*gamma1)):
> twopsi:=expand(subs(csolns2,2*psi1)):
> twotheta:=expand(subs(csolns2,2*theta1)):
> eqRcase1:=R1=lambda^2:
> eqTcase1:=T1=ck3*mu^2:
> eqScase1:=S1=lambda*mu:
> eqRcase2:=R1=(1+I)*lambda^2:
> eqTcase2:=T1=ck3*(1+I)*mu^2:
> eqScase2:=S1=(1+I)*lambda*mu:
> eqRcase3:=R1=I*lambda^2:
> eqTcase3:=T1=ck3*I*mu^2:
> eqScase3:=S1=I*lambda*mu:
> eqRcase4:=R1=I*(1+I)*lambda^2:
> eqTcase4:=T1=ck3*I*(1+I)*mu^2:
> eqScase4:=S1=I*(1+I)*lambda*mu:
```

```
> csolnscase1:=expand(subs([eqRcase1,eqScase1,eqTcase1],csolns2));
> csolnscase2:=expand(subs([eqRcase2,eqScase2,eqTcase2],csolns2));
> csolnscase3:=expand(subs([eqRcase3,eqScase3,eqTcase3],csolns2));
> csolnscase4:=expand(subs([eqRcase4,eqScase4,eqTcase4],csolns2));
```

```
> eqlambda:=lambda=x+I*y:
> eqmu:=mu=w+I*z:
> csolnscase1:=expand(subs([eqlambda,eqmu],csolnscase1));
> csolnscase2:=expand(subs([eqlambda,eqmu],csolnscase2));
> csolnscase3:=expand(subs([eqlambda,eqmu],csolnscase3));
> csolnscase4:=expand(subs([eqlambda,eqmu],csolnscase4));
```

7.2 To remove fractional coefficients, work with 2*psi and 2*theta

```
> twopsicase1:=subs(csolnscase1,2*psi1);
> twothetacase1:=subs(csolnscase1,2*theta1);
> twopsicase2:=subs(csolnscase2,2*psi1);
> twothetacase2:=subs(csolnscase2,2*theta1);
> twopsicase3:=subs(csolnscase3,2*psi1);
> twothetacase3:=subs(csolnscase3,2*theta1);
> twopsicase4:=subs(csolnscase4,2*psi1);
> twothetacase4:=subs(csolnscase4,2*theta1);
```

7.3 psi and theta are (rational) integers, so have imaginary parts equal to zero

```
> assume(x,real):
> assume(y,real):
> assume(w,real):
> assume(z,real):
> F1case1:=Re(twothetacase1);
> F1case2:=Re(twothetacase2);
> F1case3:=Re(twothetacase3);
> F1case4:=Re(twothetacase4);
> F2case1:=Im(twothetacase1);
> F2case2:=Im(twothetacase2);
> F2case3:=Im(twothetacase3);
```

```
> F2case4:=Im(twothetacase4);
> G1case1:=Re(twopsicase1);
> G1case2:=Re(twopsicase2);
> G1case3:=Re(twopsicase3);
> G1case4:=Re(twopsicase4);
> G2case1:=Im(twopsicase1);
> G2case2:=Im(twopsicase2);
> G2case3:=Im(twopsicase3);
> G2case4:=Im(twopsicase4);
> case1flag:=false:
> case2flag:=false:
> case3flag:=false:
> case4flag:=false:
> for x1 from 0 to 3 do
> for y1 from 0 to 3 do
> for w1 from 0 to 3 do
> for z1 from 0 to 3 do
> xa:=x1 mod 2:
> ya:=y1 mod 2:
> wa:=w1 mod 2:
> za:=z1 mod 2:
> if (xa<>0 or ya<>0 or wa<>0 or za<>0) then
> F1c1mod:=subs([x=x1,y=y1,w=w1,z=z1],F1case1) mod 4:
> F2c1mod:=subs([x=x1,y=y1,w=w1,z=z1],F2case1) mod 4:
> G1c1mod:=subs([x=x1,y=y1,w=w1,z=z1],G1case1) mod 4:
> G2c1mod:=subs([x=x1,y=y1,w=w1,z=z1],G2case1) mod 4:
> F1c2mod:=subs([x=x1,y=y1,w=w1,z=z1],F1case2) mod 4:
> F2c2mod:=subs([x=x1,y=y1,w=w1,z=z1],F2case2) mod 4:
> G1c2mod:=subs([x=x1,y=y1,w=w1,z=z1],G1case2) mod 4:
> G2c2mod:=subs([x=x1,y=y1,w=w1,z=z1],G2case2) mod 4:
> F1c3mod:=subs([x=x1,y=y1,w=w1,z=z1],F1case3) mod 4:
> F2c3mod:=subs([x=x1,y=y1,w=w1,z=z1],F2case3) mod 4:
> G1c3mod:=subs([x=x1,y=y1,w=w1,z=z1],G1case3) mod 4:
> G2c3mod:=subs([x=x1,y=y1,w=w1,z=z1],G2case3) mod 4:
```

```
> F1c4mod:=subs([x=x1,y=y1,w=w1,z=z1],F1case4) mod 4:
> F2c4mod:=subs([x=x1,y=y1,w=w1,z=z1],F2case4) mod 4:
> G1c4mod:=subs([x=x1,y=y1,w=w1,z=z1],G1case4) mod 4:
> G2c4mod:=subs([x=x1,y=y1,w=w1,z=z1],G2case4) mod 4:
```

Check case1

```
> if (F2c1mod=0 and G2c1mod=0 and twothetamod4=F1c1mod
and twopsimod4=G1c1mod) then
> case1flag:=true:
> fi:
```

Check case2

```
> if (F2c2mod=0 and G2c2mod=0 and twothetamod4=F1c2mod
and twopsimod4=G1c2mod) then
> case2flag:=true:
> fi:
```

Check case3

```
> if (F2c3mod=0 and G2c3mod=0 and twothetamod4=F1c3mod
and twopsimod4=G1c3mod) then
> case3flag:=true:
> fi:
```

Check case4

```
> if (F2c4mod=0 and G2c4mod=0 and twothetamod4=F1c4mod
and twopsimod4=G1c4mod) then
> case4flag:=true:
> fi:
> fi:
> od:
> od:
> od;
> if (case1flag=false and case2flag=false and case3flag=false
and case4flag=false) then
> printf(' 2-adic check 2 failed'):
```

```
> else
```

Step 8. Having found the equations, save everything to disk.

8.1 Save equations to use in improved searches

```
> read(‘heights.txt‘):
> box:=ceil(exp(ht/16));

> filename1:=cat(‘equations‘,p,‘.txt‘):
> if case1flag=true then
> F2:=F2case1:
> G2:=G2case1:
> elif case2flag=true then
> F2:=F2case2:
> G2:=G2case2:
> elif case3flag=true then
> F2:=F2case3:
> G2:=G2case3:
> else
> F2:=F2case4:
> G2:=G2case4:
> fi:
> endtime:=time():
> Q1:=subs([x='x[1]',y='x[2]',w='x[3]',z='x[4]'],F2):
> Q2:=subs([x='x[1]',y='x[2]',w='x[3]',z='x[4]'],G2):
> save Q1,Q2,box,filename1:
```

8.2 Save intermediate equations to use in backwards substitutions

```
> filename2:=cat(‘intermediate‘,p,‘.txt‘):
> R:='R':
> T:='T':
> S:='S':
> theta:='theta':
```

```
> psi:='psi':
> phi:='phi':
> b:='b':
> l:='l':
> m:='m':
> lambda:='x'+I*'y':
> mu:='w'+I*'z':
> if case1flag=true then
> R:=subs(lambda='x'+I*'y',rhs(eqRcase1)):
> T:=subs(mu='w'+I*'z',rhs(eqTcase1)):
> S:=subs([lambda='x'+I*'y',mu='w'+I*'z'],rhs(eqScase1)):
> elif case2flag=true then
> R:=subs(lambda='x'+I*'y',rhs(eqRcase2)):
> T:=subs(mu='w'+I*'z',rhs(eqTcase2)):
> S:=subs([lambda='x'+I*'y',mu='w'+I*'z'],rhs(eqScase2)):
> elif case3flag=true then
> R:=subs(lambda='x'+I*'y',rhs(eqRcase3)):
> T:=subs(mu='w'+I*'z',rhs(eqTcase3)):
> S:=subs([lambda='x'+I*'y',mu='w'+I*'z'],rhs(eqScase3)):
> else
> R:=subs(lambda='x'+I*'y',rhs(eqRcase4)):
> T:=subs(mu='w'+I*'z',rhs(eqTcase4)):
> S:=subs([lambda='x'+I*'y',mu='w'+I*'z'],rhs(eqScase4)):
> fi:
> theta:=subs(csolns2,theta1):
> theta:=subs([R1='R',S1='S',T1='T'],theta):
> psi:=subs(csolns2,psi1):
> psi:=subs([R1='R',S1='S',T1='T'],psi):
> phi:='psi'-thetamultiple*'theta':
> if plusflag=true then
> b:=subs(solnsplus,'b'):
> l:=subs(solnsplus,'l'):
> m:=subs(solnsplus,'m'):
> else
```

```
> b:=subs(solnsminus,'b'):
> l:=subs(solnsminus,'l'):
> m:=subs(solnsminus,'m'):
> fi:
> timetaken:=endtime-starttime:
> save lambda,mu,p,u,v,b,l,m,timetaken,filename2:
> fi:
> fi:
> fi:
> fi:
```

Appendix B

p6sim.mws

We wish to search for solutions of $Q1(\underline{x})=Q2(\underline{x})=0$ with $\underline{x}=(x[1],x[2],x[3],x[4])$, $x[i]$ integers in the region abs($x[i]$) <= N.

```
> restart:
> with(linalg):
> readlib(lattice):
```

Which Bremner and Cassels equation is this run for?

```
> P:=16421;
```

Read in equations and search range for the prime entered above.

```
> eqnfile:=cat(‘equations‘,P,‘.txt‘):
> read (eqnfile);
```

Start the clock.

```
> starttime:=time():
```

Pre-compute gradient vectors.

```
> gradQ1:=unapply(convert(grad(Q1, [x[1], x[2], x[3], x[4]]),
list), x):
> gradQ2:=unapply(convert(grad(Q2, [x[1], x[2], x[3], x[4]]),
list), x):
> G1:=hessian(Q1,[x[1],x[2],x[3],x[4]]):
> G2:=hessian(Q2,[x[1],x[2],x[3],x[4]]):
```

Pre-compute Taylor series expansions.

```
> Q1func:=unapply(Q1,[x[1],x[2],x[3],x[4]]):
> Q2func:=unapply(Q2,[x[1],x[2],x[3],x[4]]):
> tayQ11:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ12:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
> tayQ13:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 3), a, b):
> tayQ21:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ22:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
> tayQ23:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 3), a, b):
```

Declare vectors to be used in calculations.

```
> zvect:=[0,z[2],z[3],z[4]]:
> zdvect:=[0,zd[2],zd[3],zd[4]]:
> zddvect:=[0,zdd[2],zdd[3],zdd[4]]:
> tvect:=[0,t[2],t[3],t[4]]:
> mdvect:=[0,md[2],md[3],md[4]]:
```

Choose a prime *p* of good reduction of correct size.

```
> startpoint:=evalf(2^(5/6)*N^(1/3)):
> pfound:=false:
> for i from ceil(startpoint) while  not pfound do
> if isprime(i) then
> p:=i:
```

```
> pfound:=true:
> fi:
> od:
> p:=p; #to display
```

Step 1. Find all solutions of $Q1(\underline{x})=Q2(\underline{x})=0 \pmod p$ with $x[1]=1$. There will be $O(p)$ solutions.

1.1 Eliminate $x[4]$ so that we have to solve $F(x[1],x[2],x[3])=0 \pmod p$ for some quartic F.

```
> elim:=eliminate({Q1,Q2},x[4]):
> x4soln:=elim[1][1]:
> F:=elim[2][1]:
```

1.2 Substitute $x[1]=1$ in F and reduce modulo p.

```
> Fp:=subs(x[1]=1,F) mod p:
```

1.3 Find all solutions $\underline{x}$, and determine the corresponding $x[4]$.

```
> xcount:=-1:
> for xi from 0 to p-1 do
> x3s[xi]:=Roots(subs(x[2]=xi,Fp)) mod p;
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=1,x[2]=xi,x[3]=xj},rhs(x4soln)) mod p:
> Q1subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q1) mod p:
> Q2subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q2) mod p:
> if Q1subs=0 and Q2subs=0 then
> xcount:=xcount+1:
> xsoln[xcount]:=[1,xi,xj,xk]:
```

Step 2. Now lift each solution $\underline{x}$ to a set of p solutions $\underline{y}=\underline{x}+p\underline{z} \pmod{p\text{^}2}$ with $y[1]=1$.

2.1 Set up Taylor series expansions.

```
> eq1:=tayQ12(xsoln[xcount]+p*zvect,xsoln[xcount])/p:
> eq2:=tayQ22(xsoln[xcount]+p*zvect,xsoln[xcount])/p:
```

2.2 Solve the equations to find $\underline{z}$ in terms of one of the variables.

```
> solns1:=traperror(solve({eq1,eq2}) mod p):

> if solns1<>"the modular inverse does not exist" then
> vary1:='vary1':
> for i from 1 to nops(solns1) do
> if lhs(solns1[i])=rhs(solns1[i]) then
> vary1:=rhs(solns1[i]):
> fi:
> od:
> ztemp:=subs(solns1,zvect):
```

2.3 Search for all solutions $\underline{z}$, and determine the corresponding $\underline{y}$.

```
> for ycount from 0 to p-1 do
> zsoln[xcount,ycount]:=subs(vary1=ycount,ztemp):
> ysoln[xcount,ycount]:=xsoln[xcount]+p*zsoln[xcount,ycount]
mod p^2:
```

Step 3. Lift $\underline{y}$ to an arbitrary solution $\underline{y}'=\underline{y}+p^2\underline{z}'$ (mod p^4) with y'[1]=1.

3.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(ysoln[xcount,ycount]+p^2*zdvect, ysoln[xcount,
ycount])/p^2:
> eq2:=tayQ22(ysoln[xcount,ycount]+p^2*zdvect, ysoln[xcount,
ycount])/p^2:
```

3.2 Solve the equations to find a solution $\underline{z}'$ in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns2:=traperror(solve({eq1,eq2}) mod p^2):

> if solns2<>"the modular inverse does not exist" then
> vary2:='vary2':
> for i from 1 to nops(solns2) do
> if lhs(solns2[i])=rhs(solns2[i]) then
> vary2:=rhs(solns2[i]):
> fi:
> od:
> zdtemp:=subs(solns2,zdvect):
> zdsoln[xcount,ycount]:=subs(vary2=1,zdtemp) mod p^2:
> ydsoln[xcount,ycount]:=ysoln[xcount,ycount]+p^2*zdsoln[xcount,
ycount] mod p^4:
```

Step 4. Lift *y'* to an arbitrary solution *y"*=*y'*+*p*^4*z"* (mod *p*^8) with *y'*[1]=1.

4.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(ydsoln[xcount,ycount]+p^4*zddvect, ydsoln[xcount,
ycount])/p^4:
> eq2:=tayQ22(ydsoln[xcount,ycount]+p^4*zddvect, ydsoln[xcount,
ycount])/p^4:
```

4.2 Solve the equations to find a solution *z"* in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns3:=traperror(solve({eq1,eq2}) mod p^4):

> if solns3<>"the modular inverse does not exist" then
> vary3:='vary3':
> for i from 1 to nops(solns3) do
> if lhs(solns3[i])=rhs(solns3[i]) then
> vary3:=rhs(solns3[i]):
> fi:
> od:
> zddtemp:=subs(solns3,zddvect):
> zddsoln[xcount,ycount]:=subs(vary3=1,zddtemp):
```

```
> yddsoln[xcount,ycount]:=ydsoln[xcount,ycount] + p^4*zddsoln[xcount,
ycount] mod p^8:
```

Step 5. Reduce *y"* to a solution *y"'* (mod *p*^6).

```
> ydddsoln[xcount,ycount]:=yddsoln[xcount,ycount] mod p^6:
```

Step 6. Let *w* = lambda *y"'* + *p*^2*t*. Find a lattice basis for *t*.

6.1 Set up the Taylor series expansions.

```
> w:=evalm(evalm(lambda*ydddsoln[xcount,ycount]) + evalm(p^2*tvect)):
> temp1_1:=tayQ11(w,ydddsoln[xcount,ycount]):
> temp1_2:=tayQ12(w,ydddsoln[xcount,ycount]):
> eq1:=simplify(temp1_2-temp1_1)/p^2:
> temp2_1:=tayQ21(w,ydddsoln[xcount,ycount]):
> temp2_2:=tayQ22(w,ydddsoln[xcount,ycount]):
> eq2:=simplify(temp2_2-temp2_1)/p^2:
```

6.2 Solve the equations to find a basis vector *u* for the solution space modulo *p*^2.

```
> solns4:=traperror(solve({eq1,eq2}) mod p^4):

> if solns4<>"the modular inverse does not exist" then
> vary4:='vary4':
> for i from 1 to nops(solns4) do
> if lhs(solns4[i])=rhs(solns4[i]) then
> if lhs(solns4[i])<>lambda then
> vary4:=rhs(solns4[i]):
> fi:
> fi:
> od:
> ttemp:=subs(solns4,tvect):
> usoln:=subs(vary4=1,ttemp) mod p^4:
```

```
> use_r1:=false:use_r2:=false:use_r3:=false:
> if vary4=t[2] then
> use_r1:=true:
> elif vary4=t[3] then
> use_r2:=true:
> elif vary4=t[4] then
> use_r3:=true:
> fi:
```

Step 7. Now find the other vector in the lattice basis, *m'*.

7.1 Set up the equation.

```
> r1:=evalm(usoln&*G1&*usoln):
> r2:=evalm(usoln&*G2&*usoln):
> eq:=r2*dotprod(mdvect,gradQ1(ydddsoln[xcount,ycount]))
> - r1*dotprod(mdvect,gradQ2(ydddsoln[xcount,ycount])):

> if use_r1=true then
> eq:=subs(md[2]=0,eq):
> elif use_r2=true then
> eq:=subs(md[3]=0,eq):
> elif use_r3=true then
> eq:=subs(md[4]=0,eq):
> fi:
```

7.2 Solve the equation to find ***m'***.

```
> solns5:=traperror(solve(eq) mod p^2):

> if solns5<>"the modular inverse does not exist" then
> vary5:='vary5':
> for i from 1 to nops(solns5) do
> if lhs(solns5[i])=rhs(solns5[i]) then
> vary5:=rhs(solns5[i]):
> else
```

```
> c1:=coeffs(rhs(solns5[i])):
> fi:
> od:
```

Step 8. Find an LLL-reduced lattice basis for $\underline{w}$.

8.1 Set up the lattice basis.

```
> if use_r1=true then
> if vary5=md[4] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,0,p^4*c1,p^4],
[0,0,p^6,0]]:
> elif vary5=md[3] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,0,p^4,p^4*c1],
[0,0,0,p^6]]:
> fi:
> elif use_r2=true then
> if vary5=md[4] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,p^4*c1,0,p^4],
[0,p^6,0,0]]:
> elif vary5=md[2] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,p^4,0,p^4*c1],
[0,0,0,p^6]]:
> fi:
> elif use_r3=true then
> if vary5=md[2] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,p^4,p^4*c1,0],
[0,0,p^6,0]]:
> elif vary5=md[3] then
> latticebasis:=[ydddsoln[xcount,ycount], p^2*usoln, [0,p^4*c1,p^4,0],
[0,p^6,0,0]]:
> fi:
> fi:
```

8.2 LLL-reduce the basis.

```
> redvect:=traperror(lattice(latticebasis, 'integer')):
> if redvect<>"the ivectors are linearly dependant" then
> rv1:=redvect[1]:rv2:=redvect[2]:rv3:=redvect[3]:
```

Step 9. Express $\underline{w}$ in terms of the lattice basis vectors.

```
> s:={x[1]=v[1]*rv1[1]+v[2]*rv2[1]+v[3]*rv3[1],
> x[2]=v[1]*rv1[2]+v[2]*rv2[2]+v[3]*rv3[2],
> x[3]=v[1]*rv1[3]+v[2]*rv2[3]+v[3]*rv3[3],
> x[4]=v[1]*rv1[4]+v[2]*rv2[4]+v[3]*rv3[4]}:
> Q1w:=subs(s,Q1):
> Q2w:=subs(s,Q2):
> Q1w:=simplify(Q1w):
> Q2w:=simplify(Q2w):
```

Step 10. Solve these equations to find *v*[1], *v*[2] and *v*[3].

```
> isolns:=isolve({Q1w,Q2w}):
> if isolns<>NULL then
> w:=v[1]*rv1+v[2]*rv2+v[3]*rv3:
> w:=subs(isolns,w):
> w:=subs(_Z1=1,w):
> w:=w/igcd(w[1],w[2],w[3],w[4]):
> if norm(w)<=N then
> print(‘SOLUTION FOUND‘, w, time()-starttime):
> fi:
> fi:  #if isolns<>NULL then
> fi:  #if redvect<>"the ivectors are linearly dependant" then
> fi:  #if solns5<>"the modular inverse does not exist" then
> fi:  #if solns4<>"the modular inverse does not exist" then
> fi:  #if solns3<>"the modular inverse does not exist" then
> fi:  #if solns2<>"the modular inverse does not exist" then
> od:  #for ycount from 0 to p-1 do
> fi:  #if solns1<>"the modular inverse does not exist" then
```

```
> fi:  #if Q1subs=0 and Q2subs=0 then
> od:  #for counter from 1 to nops(x3s[xi]) do
> fi:  #if x3s[xi]<>[] then
> od:  #for xi from 0 to p-1 do
```

Stop the clock

```
> stoptime:=time():
> totaltime:=stoptime-starttime;
```

Appendix C

p4quart.mws

We wish to search for solutions of $Q1(\underline{x})=Q2(\underline{x})=0$ with $\underline{x}=(x[1],x[2],x[3],x[4])$, $x[i]$ integers in the region $|x[i]| <= N$. We first eliminate one variable and work with $F(\underline{x})=0$ with $\underline{x}=(x[1],x[2],x[3])$. The corresponding $x[4]$ is then found.

```
> restart:
> with(linalg):
> readlib(lattice):
```

Which Bremner and Cassels equation is this run for?

```
> P:=16421;
```

Read in equations and search range for the prime entered above.

```
> eqnfile:=cat('equations',P,'.txt'):
> read (eqnfile);
```

Start the clock.

```
> starttime:=time():
```

Pre-compute gradient vectors.

```
> gradF:=unapply(convert(grad(F,[x[1],x[2],x[3]]),list),x):
> G1:=hessian(F,[x[1],x[2],x[3]]):
```

Pre-compute Taylor series expansions.

```
> Ffunc:=unapply(F,[x[1],x[2],x[3]]):
> tayF1:=unapply(mtaylor(Ffunc(a[1], a[2], a[3]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3]], 1), a, b):
> tayF2:=unapply(mtaylor(Ffunc(a[1], a[2], a[3]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3]], 2), a, b):
```

Declare vectors to be used in calculations.

```
> zvect:=[0,z[2],z[3]]:
> zdvect:=[0,zd[2],zd[3]]:
> zddvect:=[0,zdd[2],zdd[3]]:
> tvect:=[0,t[2],t[3]]:
> mdvect:=[0,md[2],md[3]]:
```

Choose a prime *p* of correct size.

```
> startpoint:=evalf(12^(1/4)*box^(1/2)):
> pfound:=false:
> for i from ceil(startpoint) while  not pfound do
> if isprime(i) then
> p:=i:
> pfound:=true:
> fi:
> od:
> p:=p;
```

Eliminate *x*[4] so that we have to solve $F(x[1],x[2],x[3])=0 \pmod{p}$ for some quartic *F*.

```
> elim:=eliminate({Q1,Q2},x[4]):
> x4soln:=rhs(elim[1][1]):
> F:=elim[2][1]:
```

Step 1. Find all solutions of $F(\underline{x})=0 \pmod p$ with $x[1]=1$. There will be $O(p)$ solutions.

1.1 Substitute $x[1]=1$ in F and reduce modulo p

```
> Ftemp:=subs(x[1]=1,F) mod p:
```

1.2 Search for all solutions $\underline{x}$.

```
> xcount:=-1:
> for xi from 0 to p-1 do
> x3s[xi]:=Roots(subs(x[2]=xi,Ftemp)) mod p:
> if x3s<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xcount:=xcount+1;
> xsoln[xcount]:=[1,xi,x3s[xi][counter][1]]:
```

Step 2. Now lift each solution $\underline{x}$ to a set of p solutions $\underline{y}=\underline{x}+p\underline{z} \pmod{p\text{^}2}$ with $y[1]=1$.

2.1 Set up the Taylor series expansion.

```
> eq1:=tayF2(xsoln[xcount]+p*zvect,xsoln[xcount])/p:
```

2.2 Solve the equation to find $\underline{z}$ in terms of one variable.

```
> solns1:=traperror(solve(eq1) mod p):

> if solns1<>"the modular inverse does not exist" then
> vary1:='vary1':
> for i from 1 to nops(solns1) do
> if lhs(solns1[i])=rhs(solns1[i]) then
```

```
> vary1:=rhs(solns1[i]):
> fi:
> od:
> ztemp:=subs(solns1,zvect):
```

2.3 Search for all solutions $\underline{z}$, and determine the corresponding $\underline{y}$.

```
> for ycount from 0 to p-1 do
> zsoln[xcount,ycount]:=subs(vary1=ycount,ztemp):
> ysoln[xcount,ycount]:=xsoln[xcount]+p*zsoln[xcount, ycount]
mod p^2:
```

Step 3. Lift $\underline{y}$ to an arbitrary solution $\underline{y}'=\underline{y}+p^2\underline{z}'$ (mod p^4) with $y'[1]=1$.

3.1 Set up the Taylor series expansion.

```
> eq1:=tayF2(ysoln[xcount,ycount]+p^2*zdvect, ysoln[xcount,
ycount])/p^2:
```

3.2 Solve the equation to find a solution $\underline{z}'$ in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns2:=traperror(solve(eq1) mod p^2):

> if solns2<>"the modular inverse does not exist" then
> vary2:='vary2':
> for i from 1 to nops(solns2) do
> if lhs(solns2[i])=rhs(solns2[i]) then
> vary2:=rhs(solns2[i]):
> fi:
> od:
> zdtemp:=subs(solns2,zdvect):
> zdsoln[xcount,ycount]:=subs(vary2=1,zdtemp):
> ydsoln[xcount,ycount]:=ysoln[xcount,ycount]+p^2*zdsoln[xcount,
ycount] mod p^4;
```

Step 4. Let $\underline{w}$ = lambda $\underline{y}'$ + p^2 $\underline{t}$. Find a lattice basis for $\underline{t}$.

4.1 Set up the Taylor series expansion.

```
> w:=evalm(evalm(lambda*ydsoln[xcount,ycount])+evalm(p^2*tvect)):
> temp1:=tayF1(w,ydsoln[xcount,ycount]):
> temp2:=tayF2(w,ydsoln[xcount,ycount]):
> eq1:=simplify(temp2-temp1)/p^2 mod p^2;
```

4.2 There is no need to solve this equation as the solution is already "known".

```
> cofa:=coeff(eq1,t[2]):
> cofb:=coeff(eq1,t[3]):
> vectcontflag:=false:
> if cofa<>0 then
> if cofb=0 then
> vect:=[0,0,1]:
> else
> vect:=[0,cofb,-cofa]:
> fi:
> vectcontflag:=true:
> elif cofb<>0 then
> vect:=[0,1,0]:
> vectcontflag:=true:
> fi:

> if vectcontflag=true then
> use_r1:=false:  use_r2:=false:
> if vect[2] mod p <> 0 then
> use_r1:=true:
> vect:=vect/vect[2] mod p^2:
> elif vect[3] mod p <> 0 then
> use_r2:=true:
> vect:=vect/vect[3] mod p^2:
> fi:
```

Step 5. Find an LLL-reduced lattice basis for *w*.

5.1 Set up the lattice basis.

```
> if use_r2=true then
> latticebasis:=[ydsoln[xcount,ycount],p^2*vect,[0,p^4,0]]:
> elif use_r1=true then
> latticebasis:=[ydsoln[xcount,ycount],p^2*vect,[0,0,p^4]]:
> fi:
```

5.2 LLL-reduce the basis.

```
> redvect:=traperror(lattice(latticebasis, 'integer')):
> if redvect<>"the vectors are linearly dependent" then
> rv1:=redvect[1]:  rv2:=redvect[2]:
```

Step 6. Express *w* in terms of the lattice basis vectors.

```
> s:={x[1]=v[1]*rv1[1]+v[2]*rv2[1], x[2]=v[1]*rv1[2]+v[2]*rv2[2],
> x[3]=v[1]*rv1[3]+v[2]*rv2[3]}:
> Fw:=simplify(subs(s,F));
```

Step 7. Solve this equation to find *v*[1] and *v*[2].

```
> isoln:=isolve(Fw):
> if isoln<>NULL then
> w:=v[1]*rv1+v[2]*rv2:
> w:=subs(isoln,w):
> w:=subs(_Z1=1,w):
> w:=w/igcd(w[1],w[2],w[3]):
> w4soln:=subs({x[1]=w[1],x[2]=w[2],x[3]=w[3]},x4soln):
> if type(w4soln,integer)=true then
> if max(norm(w),w4soln)<=N then
> wsoln:=[w[1],w[2],w[3],w4soln]:
> print('SOLUTION FOUND',wsoln,time()-starttime):
> fi:
> fi:  #if type (w4soln,integer) then
```

```
> fi:  #if isoln<>NULL then
> fi:  #if redvect<>"the vectors are linearly dependent" then
> fi:  #if vectcontflag=true then
> fi:  #if solns2<>"the modular inverse does not exist" then
> od:  #for ycount from 0 to p-1 do
> fi:  #if solns1<>"the modular inverse does not exist" then
> od:  #for counter from 1 to nops(x3s[xi]) do
> fi:  #if x3s<>[] then
> od:  #for xi from 0 to p-1 do

> stoptime:=time():
> totaltime:=stoptime-starttime;
```

Appendix D

p4sim.mws

We wish to search for solutions of $Q1(\underline{x})=Q2(\underline{x})=0$ with $\underline{x}=(x[1],x[2],x[3],x[4])$, $x[i]$ integers in the region $|x[i]| <= N$.

```
> restart:
> with(linalg):
> readlib(lattice):
```

Which Bremner and Cassels equation is this run for?

```
> P:=16421;
```

Read in equations and search range for the prime entered above.

```
> eqnfile:=cat(‘equations‘,P,‘.txt‘);
> read (eqnfile);
```

Start the clock.

```
> starttime:=time():
```

Pre-compute gradient vectors.

```
> gradQ1:=unapply(convert(grad(Q1,[x[1],x[2],x[3],x[4]]), list),
x):
> gradQ2:=unapply(convert(grad(Q2,[x[1],x[2],x[3],x[4]]), list),
x):
> G1:=hessian(Q1,[x[1],x[2],x[3],x[4]]):
> G2:=hessian(Q2,[x[1],x[2],x[3],x[4]]):
```

Pre-compute Taylor series expansions.

```
> Q1func:=unapply(Q1,[x[1],x[2],x[3],x[4]]):
> Q2func:=unapply(Q2,[x[1],x[2],x[3],x[4]]):
> tayQ11:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ12:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
> tayQ21:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ22:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
```

Declare vectors to be used in calculations

```
> zvect:=[0,z[2],z[3],z[4]]:
> zdvect:=[0,zd[2],zd[3],zd[4]]:
> zddvect:=[0,zdd[2],zdd[3],zdd[4]]:
> tvect:=[0,t[2],t[3],t[4]]:
> mdvect:=[0,md[2],md[3],md[4]]:
```

Choose a prime *p* of correct size.

```
> startpoint:=evalf(2*N^(2/5)):
> pfound:=false:
> for i from ceil(startpoint) while  not pfound do
> if isprime(i) then
> p:=i:
> pfound:=true:
> fi:
> od:
```

```
> p:=p;
```

Step 1. Find all solutions of $Q1(\underline{x})=Q2(\underline{x})=0 \pmod p$ with $x[1]=1$. There will be O(p) solutions.

1.1 Eliminate $x[4]$ so that we have to solve $F(x[1],x[2],x[3])=0 \pmod p$ for some quartic F.

```
> elim:=eliminate({Q1,Q2},x[4]):
> x4soln:=elim[1][1]:
> F:=elim[2][1]:
```

1.2 Substitute $x[1]=1$ in F and reduce modulo p.

```
> Fp:=subs(x[1]=1,F) mod p:
```

1.3 Find all solutions $\underline{x}$, and determine the corresponding $x[4]$.

```
> xcount:=-1:
> for xi from 0 to p-1 do
> x3s[xi]:=Roots(subs(x[2]=xi,Fp)) mod p;
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=1,x[2]=xi,x[3]=xj},rhs(x4soln)) mod p:
> Q1subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q1) mod p:
> Q2subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q2) mod p:
> if Q1subs=0 and Q2subs=0 then
> xcount:=xcount+1:
> xsoln[xcount]:=[1,xi,xj,xk]:
```

Step 2. Now lift each solution $\underline{x}$ to a set of p solutions $\underline{y}=\underline{x}+p\underline{z} \pmod{p\text{^}2}$ with $y[1]=1$.

2.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(xsoln[xcount]+p*zvect,xsoln[xcount])/p:
> eq2:=tayQ22(xsoln[xcount]+p*zvect,xsoln[xcount])/p:
```

2.2 Solve the equations to find z in terms of one of the variables.

```
> solns1:=traperror(solve({eq1,eq2}) mod p):

> if solns1<>"the modular inverse does not exist" then
> vary1:='vary1':
> for i from 1 to nops(solns1) do
> if lhs(solns1[i])=rhs(solns1[i]) then
> vary1:=rhs(solns1[i]):
> fi:
> od:
> if vary1<>'vary1' then
> ztemp:=subs(solns1,zvect);
```

2.3 Search for all solutions $\underline{z}$, and determine the corresponding $\underline{y}$.

```
> for ycount from 0 to p-1 do
> zsoln[xcount,ycount]:=subs(vary1=ycount,ztemp):
> ysoln[xcount,ycount]:=xsoln[xcount]+p*zsoln[xcount,ycount]
mod p^2:
```

Step 3. Lift $\underline{y}$ to an arbitrary solution $\underline{y}'=\underline{y}+p^2\underline{z}'$ (mod p^4) with $y'[1]=1$.

3.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(ysoln[xcount,ycount]+p^2*zdvect, ysoln[xcount,
ycount])/p^2:
> eq2:=tayQ22(ysoln[xcount,ycount]+p^2*zdvect, ysoln[xcount,
ycount])/p^2:
```

3.2 Solve the equations to find $\underline{z}'$ in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns2:=traperror(solve({eq1,eq2}) mod p^2):

> if solns2<>"the modular inverse does not exist" then
> vary2:='vary2':
> for i from 1 to nops(solns2) do
> if lhs(solns2[i])=rhs(solns2[i]) then
> vary2:=rhs(solns2[i]):
> fi:
> od:
> zdtemp:=subs(solns2,zdvect):
> zdsoln[xcount,ycount]:=subs(vary2=1,zdtemp) mod p^2:
> ydsoln[xcount,ycount]:=ysoln[xcount,ycount]+p^2*zdsoln[xcount,
ycount] mod p^4:
```

Step 4. Let $\underline{w}$ = lambda $\underline{y'}$ + *p*^2 $\underline{t}$. Find a lattice basis for $\underline{t}$.

4.1 Set up the Taylor series expansions.

```
> w:=evalm(evalm(lambda*ydsoln[xcount,ycount])+evalm(p^2*tvect)):
> temp1_1:=tayQ11(w,ydsoln[xcount,ycount]):
> temp1_2:=tayQ12(w,ydsoln[xcount,ycount]):
> eq1:=simplify(temp1_2-temp1_1)/p^2 mod p^2:
> temp2_1:=tayQ21(w,ydsoln[xcount,ycount]):
> temp2_2:=tayQ22(w,ydsoln[xcount,ycount]):
> eq2:=simplify(temp2_2-temp2_1)/p^2 mod p^2:
```

4.2 Solve the equations to find a basis vector $\underline{u}$ for the solution space modulo *p*^2.

```
> solns3:=traperror(solve({eq1,eq2}) mod p^2):

> if solns3<>"the modular inverse does not exist" then
> vary3:='vary3':
> for i from 1 to nops(solns3) do
> if lhs(solns3[i])=rhs(solns3[i]) then
> vary3:=rhs(solns3[i]):
> fi:
```

```
> od:
> ttemp:=subs(solns3,tvect):
> usoln:=subs(vary3=1,ttemp) mod p^2:

> use_r1:=false:use_r2:=false:use_r3:=false:
> if vary3=t[2] then
> use_r1:=true:
> elif vary3=t[3] then
> use_r2:=true:
> elif vary3=t[4] then
> use_r3:=true:
> fi:
```

Step 5. Find an LLL-reduced lattice basis for $\underline{w}$.

5.1 Set up the lattice basis.

```
> if use_r3=true then
> latticebasis:=[ydsoln[xcount,ycount], p^2*usoln, [0,p^4,0,0],
[0,0,p^4,0]];
> elif use_r2=true then
> latticebasis:=[ydsoln[xcount,ycount], p^2*usoln, [0,p^4,0,0],
[0,0,0,p^4]];
> elif use_r1=true then
> latticebasis:=[ydsoln[xcount,ycount], p^2*usoln, [0,0,p^4,0],
[0,0,0,p^4]];
> fi:
```

5.2 LLL-reduce the basis.

```
> redvect:=traperror(lattice(latticebasis, 'integer')):
> if redvect<>"the vectors are linearly dependent" then
> rv1:=redvect[1]:rv2:=redvect[2]:rv3:=redvect[3]:
```

Step 6. Express $\underline{w}$ in terms of the lattice basis vectors.

```
> s:={x[1]=v[1]*rv1[1]+v[2]*rv2[1]+v[3]*rv3[1],
> x[2]=v[1]*rv1[2]+v[2]*rv2[2]+v[3]*rv3[2],
> x[3]=v[1]*rv1[3]+v[2]*rv2[3]+v[3]*rv3[3],
> x[4]=v[1]*rv1[4]+v[2]*rv2[4]+v[3]*rv3[4]}:
> Q1w:=subs(s,Q1):
> Q2w:=subs(s,Q2):
> Q1w:=simplify(Q1w):
> Q2w:=simplify(Q2w):
```

Step 7. Solve these equations to find *v*[1], *v*[2] and *v*[3].

```
> isolns:=isolve({Q1w,Q2w}):
> if isolns<>NULL then
> w:=v[1]*rv1+v[2]*rv2+v[3]*rv3:
> w:=subs(isolns,w):
> w:=subs(_Z1=1,w):
> w:=w/igcd(w[1],w[2],w[3],w[4]):
> if norm(w)<=N then
> print('SOLUTION FOUND ',w,time()-starttime):
> fi:
> fi:  #if isolns<>NULL then
> fi:  #if redvect<>"the vectors are linearly dependent" then
> fi:  #if solns3<>"the modular inverse does not exist" then
> fi:  #if solns2<>"the modular inverse does not exist" then
> od:  #for ycount from 0 to p-1 do
> fi:  #if vary1<>'vary1' then
> fi:  #if solns1<>"the modular inverse does not exist" then
> fi:  #if Q1subs=0 and Q2subs=0 then
> od:  #for counter from 1 to nops(x3s[xi]) do
> fi:  #if x3s[xi]<>[] then
> od:  #for xi from 0 to p-1 do
```

Stop the clock

```
> stoptime:=time():
> totaltime:=stoptime-starttime;
```

Appendix E

composite.mws

We wish to search for solutions of $Q1(\underline{x})=Q2(\underline{x})=0$ with $\underline{x}=(x[1],x[2],x[3],x[4])$, $x[i]$ integers in the region $|x[i]| <= N$.

```
> restart:
> with(linalg):
> readlib(lattice):
```

Which Bremner and Cassels equation is this run for?

```
> P:=16421;
```

Read in equations and search range for the prime entered above.

```
> eqnfile:=cat('equations',P,'.txt'):
> read (eqnfile);
```

Start the clock.

```
> starttime:=time():
```

Choose a composite modulus.

```
> p1:=29;
> p2:=37;
> q:=p1*p2;
```

Pre-compute gradient vectors.

```
> gradQ1:=unapply(convert(grad(Q1,[x[1],x[2],x[3],x[4]]),list),x):
> gradQ2:=unapply(convert(grad(Q2,[x[1],x[2],x[3],x[4]]),list),x):
> G1:=hessian(Q1,[x[1],x[2],x[3],x[4]]):
> G2:=hessian(Q2,[x[1],x[2],x[3],x[4]]):
```

Pre-compute Taylor series expansions.

```
> Q1func:=unapply(Q1,[x[1],x[2],x[3],x[4]]):
> Q2func:=unapply(Q2,[x[1],x[2],x[3],x[4]]):
> tayQ11:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ12:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
> tayQ13:=unapply(mtaylor(Q1func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 3), a, b):
> tayQ21:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 1), a, b):
> tayQ22:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 2), a, b):
> tayQ23:=unapply(mtaylor(Q2func(a[1], a[2], a[3], a[4]), [a[1]=b[1],
a[2]=b[2], a[3]=b[3], a[4]=b[4]], 3), a, b):
```

Declare vectors to be used in calculations.

```
> zvect:=[0,z[2],z[3],z[4]]:
> zdvect:=[0,zd[2],zd[3],zd[4]]:
> zddvect:=[0,zdd[2],zdd[3],zdd[4]]:
> tvect:=[0,t[2],t[3],t[4]]:
> mdvect:=[0,md[2],md[3],md[4]]:
```

Eliminate $x[4]$ so that we have to solve $F(x[1],x[2],x[3])=0 \pmod{q}$ for some quartic F.

```
> elim:=eliminate({Q1,Q2},x[4]):
> x4soln:=elim[1][1]:
> F:=elim[2][1]:
> F1:=subs(x[1]=1,F):
> F01:=subs({x[1]=0,x[2]=1},F):
> F001:=subs({x[1]=0,x[2]=0,x[3]=1},F):
```

Step 1. Find all solutions of Q1($\underline{x}$)=Q2($\underline{x}$)=0 (mod q) with first non-zero entry equal to 1.

1.1 Search for all solutions $\underline{x}$ mod p1, and determine the corresponding x[4].

```
> xcountp1:=-1:
```

Is [1,x2,x3,x4] a solution?

```
> for xi from 0 to p1-1 do
> x3s[xi]:=Roots(subs(x[2]=xi,F1)) mod p1:
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=1,x[2]=xi,x[3]=xj},rhs(x4soln)) mod p1:
> Q1subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q1) mod p1:
> Q2subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q2) mod p1:
> if Q1subs=0 and Q2subs=0 then
> xcountp1:=xcountp1+1:
> xsolnp1[xcountp1]:=[1,xi,xj,xk]:
> fi:
> od:
> fi:
> od:
```

Is [0,1,x3,x4] a solution?

```
> x3s[xi]:=Roots(F01) mod p1:
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=0,x[2]=1,x[3]=xj},rhs(x4soln)) mod p1:
> Q1subs:=subs({x[1]=0,x[2]=1,x[3]=xj,x[4]=xk},Q1) mod p1:
> Q2subs:=subs({x[1]=0,x[2]=1,x[3]=xj,x[4]=xk},Q2) mod p1:
> if Q1subs=0 and Q2subs=0 then
> xcountp1:=xcountp1+1:
> xsolnp1[xcountp1]:=[0,1,xj,xk]:
> fi:
> od:
> fi:
```

Is [0,0,1,x4] a solution?

```
> if F001 mod p1 =0 then
> xk:=subs({x[1]=0,x[2]=0,x[3]=1},rhs(x4soln)) mod p1:
> Q1subs:=subs({x[1]=0,x[2]=0,x[3]=1,x[4]=xk},Q1) mod p1:
> Q2subs:=subs({x[1]=0,x[2]=0,x[3]=1,x[4]=xk},Q2) mod p1:
> if Q1subs=0 and Q2subs=0 then
> xcountp1:=xcountp1+1:
> xsolnp1[xcountp1]:=[0,0,1,xk]:
> fi:
> fi:
```

Is [0,0,0,1] a solution?

```
> Q1subs:=subs({x[1]=0,x[2]=0,x[3]=0,x[4]=1},Q1) mod p1:
> Q2subs:=subs({x[1]=0,x[2]=0,x[3]=0,x[4]=1},Q2) mod p1:
> if Q1subs=0 and Q2subs=0 then
> xcountp1:=xcountp1+1:
> xsolnp1[xcountp1]:=[0,0,0,1]:
> fi:
```

1.2 Search for all solutions $\underline{x}$ mod p2, and determine the corresponding $x[4]$.

```
> xcountp2:=-1:
```

Is [1,x2,x3,x4] a solution?

```
> for xi from 0 to p2-1 do
> x3s[xi]:=Roots(subs(x[2]=xi,F1)) mod p2:
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=1,x[2]=xi,x[3]=xj},rhs(x4soln)) mod p2:
> Q1subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q1) mod p2:
> Q2subs:=subs({x[1]=1,x[2]=xi,x[3]=xj,x[4]=xk},Q2) mod p2:
> if Q1subs=0 and Q2subs=0 then
> xcountp2:=xcountp2+1:
> xsolnp2[xcountp2]:=[1,xi,xj,xk]:
> fi:
> od:
> fi:
> od:
```

Is [0,1,x3,x4] a solution?

```
> x3s[xi]:=Roots(F01) mod p2:
> if x3s[xi]<>[] then
> for counter from 1 to nops(x3s[xi]) do
> xj:=x3s[xi][counter][1]:
> xk:=subs({x[1]=0,x[2]=1,x[3]=xj},rhs(x4soln)) mod p2:
> Q1subs:=subs({x[1]=0,x[2]=1,x[3]=xj,x[4]=xk},Q1) mod p2:
> Q2subs:=subs({x[1]=0,x[2]=1,x[3]=xj,x[4]=xk},Q2) mod p2:
> if Q1subs=0 and Q2subs=0 then
> xcountp2:=xcountp2+1:
> xsolnp2[xcountp2]:=[0,1,xj,xk]:
> fi:
> od:
> fi:
```

Is [0,0,1,x4] a solution?

```
> if F001 mod p2 =0 then
> xk:=subs({x[1]=0,x[2]=0,x[3]=1},rhs(x4soln)) mod p2:
> Q1subs:=subs({x[1]=0,x[2]=0,x[3]=1,x[4]=xk},Q1) mod p2:
> Q2subs:=subs({x[1]=0,x[2]=0,x[3]=1,x[4]=xk},Q2) mod p2:
> if Q1subs=0 and Q2subs=0 then
> xcountp2:=xcountp2+1:
> xsolnp2[xcountp2]:=[0,0,1,xk]:
> fi:
> fi:
```

Is [0,0,0,1] a solution?

```
> Q1subs:=subs({x[1]=0,x[2]=0,x[3]=0,x[4]=1},Q1) mod p2:
> Q2subs:=subs({x[1]=0,x[2]=0,x[3]=0,x[4]=1},Q2) mod p2:
> if Q1subs=0 and Q2subs=0 then
> xcountp2:=xcountp2+1:
> xsolnp2[xcountp2]:=[0,0,0,1]:
> fi:
```

1.3 Find constants for Chinese remainder theorem.

```
> M1:=q/p1:  M2:=q/p2:
> n1:=1/M1 mod p1:
> n2:=1/M2 mod p2:
```

1.4 Combine each solution modulo p1 with each solution modulo p2 using Chinese Remainder Theorem.

Chinese remainder theorem states that x = xp2*k*p1 + xp1*t*p2 is a solution modulo p1*p2=q

```
> xcount:=-1:
> for p1count from 0 to xcountp1 do
> for p2count from 0 to xcountp2 do
> xcount:=xcount+1:
```

```
> xsoln[xcount]:=n1*M1*xsolnp1[p1count] + n2*M2*xsolnp2[p2count]
mod q:
```

Step 2. Lift each $\underline{x}$ to an arbitrary solution $\underline{y}=\underline{x}+q\underline{z}$ (mod q^2) with y[1]=1.

2.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(xsoln[xcount]+q*zvect,xsoln[xcount])/q:
> eq2:=tayQ22(xsoln[xcount]+q*zvect,xsoln[xcount])/q:
```

2.2 Solve the equations to find a solution $\underline{z}$ in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns1:=traperror(solve({eq1,eq2}) mod q):

> if solns1<>"the modular inverse does not exist" then
> vary1:='vary1':
> for i from 1 to nops(solns1) do
> if lhs(solns1[i])=rhs(solns1[i]) then
> vary1:=rhs(solns1[i]):
> fi:
> od:
> ztemp:=subs(solns1,zvect):
> zsoln[xcount]:=subs(vary1=1,ztemp):
> ysoln[xcount]:=xsoln[xcount]+q*zsoln[xcount] mod q^2:
```

Step 3. Lift each $\underline{y}$ to an arbitrary solution $\underline{y'}=\underline{y}+q^{2*}\underline{z'}$ (mod q^4) with y'[1]=1.

3.1 Set up the Taylor series expansions.

```
> eq1:=tayQ12(ysoln[xcount]+q^2*zdvect,ysoln[xcount])/q^2:
> eq2:=tayQ22(ysoln[xcount]+q^2*zdvect,ysoln[xcount])/q^2:
```

3.2 Solve the equations to find a solution $\underline{z'}$ in terms of one of the variables. Let this variable be 1 in order to find an arbitrary solution.

```
> solns2:=traperror(solve({eq1,eq2}) mod q^2):

> if solns2<>"the modular inverse does not exist" then
> vary2:='vary2':
> for i from 1 to nops(solns2) do
> if lhs(solns2[i])=rhs(solns2[i]) then
> vary2:=rhs(solns2[i]):
> fi:
> od:
> zdtemp:=subs(solns2,zdvect):
> zdsoln[xcount]:=subs(vary2=1,zdtemp):
> ydsoln[xcount]:=ysoln[xcount]+q^2*zdsoln[xcount] mod q^4:
```

Step 4. Reduce *y'* to a solution *y"* mod *q*^3.

```
> yddsoln[xcount]:=ydsoln[xcount] mod q^3:
```

Step 5. Let *w* = lambda *y"* + *qt*. Find a lattice basis for *t*.

5.1 Set up the Taylor series expansions.

```
> w:=evalm(evalm(lambda*yddsoln[xcount])+evalm(q*tvect)):
> temp1_1:=tayQ11(w,yddsoln[xcount]):
> temp1_2:=tayQ12(w,yddsoln[xcount]):
> eq1:=simplify(temp1_2-temp1_1)/q:
> temp2_1:=tayQ21(w,yddsoln[xcount]):
> temp2_2:=tayQ22(w,yddsoln[xcount]):
> eq2:=simplify(temp2_2-temp2_1)/q:
```

5.2 Solve the equations to find a basis vector ***u*** for the solution space modulo *q*.

```
> solns3:=traperror(solve({eq1,eq2}) mod q^2):

> if solns3<>"the modular inverse does not exist" then
> vary3:='vary3':
> for i from 1 to nops(solns3) do
```

```
> if lhs(solns3[i])=rhs(solns3[i]) then
> if lhs(solns3[i])<>lambda then
> vary3:=rhs(solns3[i]):
> fi:
> fi:
> od:
> ttemp:=subs(solns3,tvect):
> usoln:=subs(vary3=1,ttemp) mod q^2:

> use_r1:=false:use_r2:=false:use_r3:=false:
> if vary3=t[2] then
> use_r1:=true:
> elif vary3=t[3] then
> use_r2:=true:
> elif vary3=t[4] then
> use_r3:=true:
> fi:
```

Step 6. Now find the other vector in the lattice basis, *m'*.

6.1 Set up the equation.

```
> r1:=evalm(usoln&*G1&*usoln):
> r2:=evalm(usoln&*G2&*usoln):
> eq:= r2*dotprod(mdvect, gradQ1(yddsoln[xcount])) - r1*dotprod(mdvect,
gradQ2(yddsoln[xcount])):

> if use_r1=true then
> eq:=subs(md[2]=0,eq):
> elif use_r2=true then
> eq:=subs(md[3]=0,eq):
> elif use_r3=true then
> eq:=subs(md[4]=0,eq):
> fi:
```

6.2 Solve the equation to find ***m'***.

```
> solns4:=traperror(solve(eq) mod q):
> if solns4<>"the modular inverse does not exist" then
> vary4:='vary4':
> for i from 1 to nops(solns4) do
> if lhs(solns4[i])=rhs(solns4[i]) then
> vary4:=rhs(solns4[i]):
> else
> c1:=coeffs(rhs(solns4[i])):
> fi:
> od:
```

Step 7. Find an LLL-reduced lattice basis for *w*.

7.1 Set up the lattice basis.

```
> if use_r1=true then
> if vary4=md[4] then
> latticebasis:=[yddsoln[xcount], q*usoln, [0,0,q^2*c1,q^2],
[0,0,q^3,0]]:
> elif vary4=md[3] then
> latticebasis:=[yddsoln[xcount], q*usoln, [0,0,q^2,q^2*c1],
[0,0,0,q^3]]:
> fi:
> elif use_r2=true then
> if vary4=md[4] then
> latticebasis:=[yddsoln[xcount], q*usoln, [0,q^2*c1,0,q^2],
[0,q^3,0,0]]:
> elif vary4=md[2] then
> latticebasis:=[yddsoln[xcount], q*usoln, [0,q^2,0,q^2*c1],
[0,0,0,q^3]]:
> fi:
> elif use_r3=true then
> if vary4=md[2] then
> latticebasis:=[yddsoln[xcount], q*usoln, [0,q^2,q^2*c1,0],
[0,0,q^3,0]]:
> elif vary4=md[3] then
```

```
> latticebasis:=[yddsoln[xcount], q*usoln, [0,q^2*c1,q^2,0],
[0,q^3,0,0]]:
> fi:
> fi:
```

7.2 LLL-reduce the basis.

```
> redvect:=traperror(lattice(latticebasis, 'integer')):
> if redvect<>"the vectors are linearly dependent" then
> rv1:=redvect[1]:rv2:=redvect[2]:rv3:=redvect[3]:
```

Step 8. Express *w* in terms of the lattice basis vectors.

```
> s:=
> {x[1]=v[1]*rv1[1]+v[2]*rv2[1]+v[3]*rv3[1],x[2]=v[1]*rv1[2]+v[2]*rv2[2
> ]+v[3]*rv3[2],x[3]=v[1]*rv1[3]+v[2]*rv2[3]+v[3]*rv3[3],x[4]=v[1]*rv1[4
> ]+v[2]*rv2[4]+v[3]*rv3[4]}:
> Q1w:=subs(s,Q1):
> Q2w:=subs(s,Q2):
> Q1w:=simplify(Q1w):
> Q2w:=simplify(Q2w):
```

Step 9. Solve these equations to find *v*[1], *v*[2] and *v*[3].

```
> isolns:=isolve({Q1w,Q2w}):
> if isolns<>NULL then
> w:=v[1]*rv1+v[2]*rv2+v[3]*rv3:
> w:=subs(isolns,w):
> w:=subs(_Z1=1,w):
> w:=w/igcd(w[1],w[2],w[3],w[4]):
> if norm(w)<=N then
> print('SOLUTION FOUND ',w,time()-starttime):
> fi:
> fi:  #if isolns<>NULL then
> fi:  #if redvect<>"the vectors are linearly dependent" then
```

```
> fi:  #if solns4<>"the modular inverse does not exist" then
> fi:  #if solns3<>"the modular inverse does not exist" then
> fi:  #if solns2<>"the modular inverse does not exist" then
> fi:  #if solns1<>"the modular inverse does not exist" then
> od:  #for p2count from 0 to xcountp2 do
> od:  #for p1count from 0 to xcountp1 do
> totaltime:=time()-starttime;
```

Bibliography

[1] A. Akhavi, Worst-case complexity of the optimal LLL-algorithm, *Latin 2000: Theoretical Informatics* **1776** (2000), 355-366.

[2] C. Batut, K. Belabas, D. Bernardi, H. Cohen, M. Olivier, *pari-gp*, See http://pari.math.u-bordeaux.fr/

[3] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves I, *Journal für die Reine und Angewandte Mathematik* **212** (1963), 7-25.

[4] B. J. Birch and H. P. F. Swinnerton-Dyer, Notes on elliptic curves II, *Journal für die Reine und Angewandte Mathematik* **218** (1965), 79-108.

[5] E. Bombieri, The Mordell conjecture revisited, *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* **17** (1990), 615-640.

[6] A. Bremner, On the equation $Y^2 = X(X^2 + p)$, In *Number Theory and Applications* (NATO ASI Series, Series C **265**), R. A. Mollin, ed., Dordrecht: Kluwer Academic Publishers, 1989, 3-23.

[7] A. Bremner and D. A. Buell, Three points of great height on elliptic curves, *Mathematics of Computation* **61** (1993), 111-115.

[8] A. Bremner and J. W. S. Cassels, On the equation $Y^2 = X(X^2 + p)$, *Mathematics of Computation* **42** (1984), 257-264.

[9] C. Breuil, B. Conrad, F. Diamond and R. Taylor, On the modularity of elliptic curves over $\mathbb{Q}$, *Journal of the American Mathematical Society* **14** (2001), 843-939

[10] H. Cohen, *A Course in Computational Algebraic Number Theory*, Graduate Texts in Mathematics **138**, Berlin: Springer-Verlag, 1993.

[11] I. Connell, *Apecs* (a Maple program for arithmetic of plane elliptic curves), 1995. See ftp://ftp.math.mcgill.ca/pub/apecs/. Maple required.

[12] J. E. Cremona, *Algorithms for Modular Elliptic Curves*, 2nd edition, Cambridge University Press, 1997.

[13] J. E. Cremona, Higher descents on elliptic curves, In *Huitièmes Rencontres Arithmétiques de Caen*, Boxall, ed. Preprint available from http://www.maths.nott.ac.uk/personal/jec/papers/index.html

[14] J. E. Cremona, Classical invariants and 2-descent on elliptic curves, *Journal of Symbolic Computation* **31** (2001), 71-87.

[15] J. E. Cremona, *heegner.gp*, See http://www.maths.nott.ac.uk/personal/jec/ftp/progs/

[16] J. E. Cremona, *mwrank*, See http://www.maths.nott.ac.uk/personal/jec/ftp/progs/

[17] H. Darmon and P. Green, Elliptic curves and class fields of real quadratic fields: algorithms and evidence, *Experimental Mathematics* **11** (2002), 37-55.

[18] N. D. Elkies, Rational points near curves and small nonzero $| x^3 - y^2 |$ via lattice reduction, In *Algorithmic Number Theory*, ANTS-I (Ithaca 1994), L.M. Adleman and M. D. Huang eds., Springer Lecture Notes in Computer Science **877** (1994), 122-133.

[19] G. Faltings, Endlichkeitssätze für abelsche Varietäten über Zahlkörpern, *Inventiones Mathematicae* **73** (1983), 349-366. English translation in *Arithmetic Geometry*, G. Cornell and J. Silverman eds., New York-Berlin: Springer-Verlag (1986), 9-27.

[20] J. Gebel, Data on Mordell's Curve, Available online from http://tnt.math.metro-u.ac.jp/simath/MORDELL/index.html.

[21] J. Gebel, A. Pethö and H. G. Zimmer, On Mordell's equation, *Compositio Mathematica* **110** (1998), 335-367.

[22] B. Gross and D. Zagier, Heegner points and derivatives of L-series, *Inventiones Mathematicae* **84** (1986), 225-320.

[23] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 5th edition, Clarendon Press, 1979.

[24] D. Hilbert and A. Hurwitz, Über die diophantischen Gleichungen vom Geschlecht Null, *Acta Mathematica* **14** (1890), 217-224.

[25] A. K. Lenstra, H. W. Lenstra and L Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen* **261** (1982), 515-534.

[26] C.-E. Lind, Untersuchungen über die rationalen Punkte der ebenen kubischen Kurven vom Geschlecht Eins, PhD thesis, University of Uppsala, 1940.

[27] E. Lutz, Sur l'equation $y^2 = x^3 - Ax - B$ dans les corps p-adic, *Journal für die Reine und Angewandte Mathematik* **177** (1937), 237-247.

[28] *Magma*, developed by the Computational Algebra Group at the University of Sydney. See http://magma.maths.usyd.edu.au/magma/.

[29] D. Masser and G. Wüstholz, Isogeny estimates for abelian varieties, and finiteness theorems, *Annals of Mathematics* **137** (1993), 459-472.

[30] B. Mazur, Modular curves and the Eisenstein ideal, *Publications Mathématiques de l'IHÉS* **47** (1977), 33-186.

[31] B. Mazur, Rational isogenies of prime degree, *Inventiones Mathematicae* **44** (1978), 129-162.

[32] J. R. Merriman, S. Siksek and N. P. Smart, Explicit 4-descents on an elliptic curve, *Acta Arithmetica* **LXXVII.4** (1996), 385-404.

[33] P. Monsky, Three constructions of rational points on $Y^2 = X^3 \pm NX$, *Mathematische Zeitschrift* **209** (1992), 445-462.

[34] L. J. Mordell, *Diophantine Equations*, Academic Press, 1969.

[35] L. J. Mordell, On the magnitude of the integer solutions of the equation $ax^2 + by^2 + cz^2 = 0$, *Journal of Number Theory* **1** (1969), 1-3.

[36] T. Nagell, Solution de quelque problemes dans la theorie arithmetiques des cubiques planes du premier genre, *Skrifter utgitt av det Norske videnskaps-akademi i Oslo. I, Matematisk-naturvidenskapelig Klasse* **1** (1935), 1-25.

[37] M. J. Razar, A relation between the two-component of the Tate-Shafarevich group and $L(1)$ for certain elliptic curves, *American Journal of Mathematics* **96** (1974), 127-144.

[38] H. Reichardt, Einige im Kleinen überall lösbare, im Grossen unlösbare diophantische Gleichungen, *Journal für die Reine und Angewandte Mathematik* **184** (1942), 12-18.

[39] H. E. Rose, *A Course in Number Theory*, Clarendon Press, 1988.

[40] H. E. Rose, On a class of elliptic curves with rank at most two, *Mathematics of Computation* **64** (1995), 1251-1265.

[41] E. S. Selmer, The diophantine equation $ax^3 + by^3 + cz^3 = 0$, *Acta Mathematica* **85** (1951), 203-362.

[42] S, Siksek, Descent on curves of genus one, PhD thesis, University of Exeter, 1995. Available online from http://www.maths.nott.ac.uk/personal/jec/theses/index.html.

[43] J. H. Silverman, *The Arithmetic of Elliptic Curves*, New York: Springer-Verlag, 1986.

[44] J. H. Silverman, The difference between the Weil height and the canonical height on elliptic curves, *Mathematics of Computation* **55** (1990), 723-743.

[45] J. H. Silverman, Computing rational points on rank 1 elliptic curves via L-series and canonical heights, *Mathematics of Computation* **68** (1999), 835-858.

[46] N. P. Smart, *The Algorithmic Resolution of Diophantine Equations*, London Mathematical Society Student Texts **41**, Cambridge University Press, 1998.

[47] M. Stoll, Explicit 4-descent on an elliptic curve, Available online from http://modular.math.washington.edu/scans/papers/stoll/4-descent.pdf.

[48] R. J. Stroeker and J. Top, On the equation $Y^2 = (X + p)(X^2 + p^2)$, *Rocky Mountain Journal of Mathematics* **24** (1994), 1135-1161.

[49] R. Taylor and A. Wiles, Ring-theoretic properties of certain Hecke algebras, *Annals of Mathematics* **141** (1995), 553-572.

[50] P. Vojta, Siegel's theorem in the compact case, *Annals of Mathematics* **133** (1991), 509-548.

[51] B. M. M. de Weger, Solving exponential diophantine equations using lattice basis reduction algorithms, *Journal of Number Theory* **26** (1987), 325-367.

[52] A. Wiles, Modular elliptic curves and Fermat's Last Theorem, *Annals of Mathematics* **141** (1995), 443-551.

[53] T. Womack, Explicit descent on elliptic curves, PhD thesis, University of Nottingham, 2003. Available online from http://www.maths.nott.ac.uk/personal/jec/theses/index.html

[54] D. Zagier, Modular points, modular curves, modular surfaces and modular forms. In *Workshop Bonn 1984 (Bonn, 1984)* (Lecture Notes in Mathematics **1111**), Berlin: Springer Verlag, 1985, 225-248.

[55] H. G. Zimmer, On the difference between the Weil height and the Neron-Tate height, *Mathematische Zeitschrift* **147** (1976), 35-51.