RANKS OF QUADRATIC TWISTS OF ELLIPTIC CURVES WITH NO ISOGENIES

CHRISTOPHER DAVIS

Abstract. A method is outlined in [El94] for constructing rational points on quadratic twists of elliptic curves. Assuming the Birch and Swinnerton-Dyer conjecture, the method also produces a way to identify curves of rank at least three among certain quadratic twists of an arbitrary elliptic curve of rank zero. In this paper, we describe a use of the GP/PARI package to carry out the construction for a number of curves. In particular, we study the frequency of curves of rank three among the quadratic twists of several rank zero curves with no non-trivial isogenies.

1. Introduction: Quadratic Twists

Working in decreasing generality, we recall briefly the interest in ranks of elliptic curves, then discuss some current knowledge concerning ranks in families of quadratic twists, and finally describe and implement a process for studying the frequency of rank 3 curves among certain quadratic twists of elliptic curves with no isogenies. We choose the specific context we do because several recent theorems and conjectures concerning rank 3 twists seem to make an unexpected distinction between those curves with isogenies and those without. (Throughout this paper, we refer to curves $E(\mathbb{Q})$ which possess only the usual "multiplication by m" endomorphisms as elliptic curves with no isogenies. Equivalently, these are the curves for which the only isogenous curves are isomorphic as well.)

The rank of an elliptic curve over $\mathbb{Q}$ is one of the most studied invariants in contemporary mathematics. Our knowledge of such ranks is miniscule compared to our knowledge of the corresponding torsion subgroups, about which few interesting questions remain. We have an abundance of open questions concerning rank, though. Most notable among them is the Birch and Swinnerton-Dyer Conjecture, but there are plenty of others. For instance, it is not known whether ranks of elliptic curves over the rational numbers are bounded.

Other questions involve ranks in particular families of elliptic curves. One such family is comprised of the quadratic twists of an elliptic curve:

Definition 1.1. For an elliptic curve $E : \{y^2 = x^3 + ax + b\}$ with $a, b \in \mathbb{Q}$, the *quadratic twist* of E by a nonzero rational number D is the elliptic curve $E_D : \{Dy^2 = x^3 + ax + b\}$.

These curves are all closely related, as, for one, they are isomorphic over $\overline{\mathbb{Q}}$, with an isomorphism given by

$$(x, y) \in E(\overline{\mathbb{Q}}) \Leftrightarrow (x, y/\sqrt{d}) \in E_d(\overline{\mathbb{Q}}).$$

Date: June 3, 2004.

Over $\mathbb{Q}$, however, their arithmetic can be wholly unrelated. For this reason, families of quadratic twists provide a productive grounds for studying arithmetic properties of elliptic curves in general. A question concerning a family of quadratic twists is, in general, much more tractable than a question concerning all elliptic curves over $\mathbb{Q}$. In particular, our questions concerning rank distribution are much easier to manage in a family of quadratic twists. One reason is that, while quadratic twists come with the ordering given by their rational D's (which we will see later can actually be restricted to squarefree integers, making them even easier to order), in general, elliptic curves over $\mathbb{Q}$ come with nothing so natural. (The conductor, though different curves may have the same conductor, is one possibility for ordering.)

Consequently, the results concerning rank distribution in families of quadratic twists are often much stronger than the results in general. In many such families, a fair amount is known. Even more is known when we allow ourselves the use of certain conjectures which are widely believed to be true. The conclusion we eventually draw in this paper relies on both the Birch and Swinnerton-Dyer Conjecture and its immediate corollary, the Parity Conjecture. A good deal of current research relies on one or both of these conjectures, as well as others.

For the purposes of summarizing some of what is known concerning rank frequencies of quadratic twists, we will borrow notation from [RS02]. Let $S(X) = \{\text{squarefree } d \in \mathbb{Z} : |d| \leq X\}$. We need only consider squarefree integers when studying quadratic twists, because square-terms can be absorbed into y, and any remaining denominators can be eliminated by multiplying the numerator by the corresponding square. Define $N_*(X) = \#\{d \in S(X) : \text{rank}(E_d) \textit{ is } *\}$ where $*$ is any property that makes sense, such as "1", "≥ 2", etc. Finally, define $D_*(E) = \lim_{X\to\infty} \frac{N_*(X)}{N(X)}$, assuming the limit exists. We are now ready to state some current results and conjectures concerning ranks of quadratic twists.

The techniques described in this paper concern finding curves of rank 3, but it is of course helpful to survey some of the known and conjectured results concerning twists of small rank in general. Assuming the Parity Conjecture, with the above notation, we have $D_{\text{even}} = D_{\text{odd}} = \frac{1}{2}$. Much bolder than this, and yet one of the most basic conjectures concerning these ranks, is the *Density Conjecture*: $D_0(E) = D_1(E) = \frac{1}{2}$, and $D_{\geq 2}(E) = 0$. In words, the Density Conjecture claims that, for any elliptic curve E over $\mathbb{Q}$, on average, half of its quadratic twists have rank 0, and half have rank 1. (The Density Conjecture would follow from the Parity Conjecture and another conjecture called the Goldfeld Conjecture, see [RS02].)

As the Density Conjecture suggests, curves of rank 0 and 1 are ubiquitous among the quadratic twists of an elliptic curve. Curves of higher rank are much rarer, and finding them often requires some cleverness. It has been conjectured that in any given family of quadratic twists, the ranks of the twists are bounded. But this conjecture is less trusted than many of the conjectures we describe (see p. 467 of [RS02]).

Recent conjectures and numerical evidence suggest that twists of rank 2 and rank 3 *aren't that* rare. Some expect that $N_{\geq 2}(X)$ and $N_{\geq 3}(X)$ both grow like $X^{3/4}$. (Again, see [RS02].) As far as what has been proved, however, the results are notably weaker. Below we write $A(X) \gg B(X)$ to indicate that there is a constant C depending only on E such that $A(X) \geq CB(X)$ for sufficiently large X. The following is a variant of what is included in **Theorem** 8.2 in [RS02]:

Theorem 1.2. *Without assuming the Parity Conjecture or any others, we have:*

(1) $N_{\geq 1}(X) \gg X^{1/2}$.
(2) *If* E *is* $y^2 = x^3 + ax + b$ *and* $ab \neq 0$, *then* $N_{\geq 2}(X) \gg X^{1/7}/\log^2(X)$.
(3) *If* E *is* $y^2 = x(x-f)(x-c^2f)$ *or* $y^2 = x(x-f)(x+2c^2f)$ *with* $c, f \in \mathbb{Q}$, *then* $N_{\geq 3}(X) \gg X^{1/6}$.

Assuming the Parity Conjecture, we obtain moreover:

(3′) *If* E *is as in 3, and in many other cases as well,* $N_{\geq 3}(X) \gg X^{1/3}$.

The above discussion and theorem should make apparent the divisions between what we know, what believable conjectures imply, and what certain evidence suggests. Restricting to the special elliptic curves considered in (3) above, we have that $N_{\geq 3}(X) \gg X^{1/6}$, $X^{1/3}$, or $X^{3/4}$ in the three cases, respectively.

The proof relies on an important construction which uses different sorts of twists from the ones we've described thus far. Consider as usual an elliptic curve $E : y^2 = f(x)$, defined over $\mathbb{Q}$. Although it may seem more natural to take $x, y \in \mathbb{Q}, \mathbb{R}, \mathbb{C}$, etc., we can just as well look for solutions in any field which contains $\mathbb{Q}$ as a subfield. In particular, we can look for solutions in the rational function field $\mathbb{Q}(t)$. To this end, we define a twist by a rational function $g(t)$:

$$E_{g(t)} : g(t)y^2 = f(x),$$

where a solution (x, y) will be a pair of rational functions in t satisfying the equation. A single such twist with rank r (as an elliptic curve over $\mathbb{Q}(t)$) will in fact produce infinitely many quadratic twists with rank r *over* $\mathbb{Q}$:

Given r independent points $P_1(t), \ldots, P_r(t)$, for all but finitely many $t_0 \in \mathbb{Q}$, mere plugging in will yield r independent points in $E_{g(t_0)}(\mathbb{Q})$. (We need to avoid all t_0's which are roots of $g(t)$, roots of the denominator of one of the $P_i(t)$'s, plus a certain finite set for which the points $P_1(t_0), \ldots, P_r(t_0)$ are no longer independent, see [Si83]. No other t's need be avoided.) Although in general $E_{g(t_0)}$ will be a twist by a non-squarefree integer, it will be isomorphic to a twist by a squarefree integer, and so that yields a twist of rank at least r. Without further argument, though, there is nothing to prevent the function of t, squarefree$(g(t))$, from taking only finitely many values. (We use squarefree(x) to denote the function which sends $x \in \mathbb{Q}$ to its corresponding squarefree integer, in the manner described above.)

Theorem 1.3. *Let* $g(t)$ *denote a squarefree polynomial of degree greater than* 4. *Then only finitely many* t*'s correspond to the same value of squarefree*$(g(t))$.

Before proceeding to the proof, we make a few comments. First, the restriction that $g(t) \in \mathbb{Q}(t)$ be a squarefree polynomial is not really a restriction, just as above where we saw that instead of considering twists by rational numbers, we could restrict to the case of twists by squarefree integers. Second, we note that the theorem implies that knowing the rank of a twist by a single $g(t)$ enables us to put that rank as a lower bound for infinitely many quadratic twists by squarefree integers.

A stronger result given in [ST95] is actually quite deep, and we describe it informally below. We will still sketch the argument for our special case, because that case is already powerful, and its proof is interesting (and very different from the proof given in [ST95]).

Proof. The question of how many different t's lead to the same squarefree part of $g(t)$ equates to counting the number of different t's appearing among the rational solutions (s,t) to the curve

$$g(t) = s^2 d,$$

where d is the fixed number which we think of as the squarefree part. In general, this will be a hyper-elliptic curve of genus $\lfloor \frac{\deg\,(g)-1}{2} \rfloor$. This is where we restrict to the special case deg $g > 4$, because in this case we can use the Mordell Conjecture, which states that the curve will have only finitely many solutions, and in particular only finitely many different t's appearing in the solutions. Thus, the function squarefree$(g(t))$ must indeed attain infinitely many values.

□

The general result, in addition to working for other polynomials $g(t)$, is also stronger than what we have shown. We showed only that there corresponded infinitely many squarefree parts, whereas the result in [ST95] will say something about what proportion of the possible squarefree values in a certain range are actually attained. Our result is also less than optimal because our entire method of finding averages requires that we observe twists over certain ranges of numbers, and the method we described above relies on the function squarefree$(g(t))$, which could grow unpredictably, even compared with the growth of $g(t)$. With these deficiencies in mind, we will describe the general result.

Although we are interested in values of $g(t)$ on rational numbers, it is helpful to consider it as a function of two variables x, y, where $t = x/y$, with x and y integers. We are allowed to multiply by squares without affecting the corresponding twist, so we multiply by the least even power of y which clears the denominators. For instance, if we begin with the function $g(t) = t^3 + 2t + 4$, we consider instead the corresponding binary form $G(x,y) = yx^3 + 2y^3x + 4y^4$.

The general result concerns only the squarefree values of $G(x,y)$, and not the values of squarefree$(G(x,y))$. Thus, its growth is significantly more predictable. Furthermore, the general result assures us that, under the restriction that $G(x,y)$ has degree at least 3, a healthy portion of all possible squarefree values appear. The amount is on the order of $X^{1/n}$ for some value n depending only on $G(x,y)$. [ST95] should be consulted for more details and precise statements.

These results combine to produce the type of bounds given earlier in Theorem 1.2. The $X^{1/3}$ estimate given there concerning the proportion of twists with rank 3 is higher than certain people expected. (And the $X^{3/4}$ conjecture mentioned above is significantly higher.) A competing conjecture[1] concerning the proportion of twists with rank 3 for general elliptic curves $E(\mathbb{Q})$ (so, not just for those curves considered in Theorem 1.2, Parts 3 and 3′) predicted that growth of $N_3(X)$ would typically be on the order of $X^{1/4}$.

For a large class of examples, at least assuming the Parity Conjecture, this conjecture has already been bettered. We recall some of the argument, lifting two theorems directly and without proof from [RS01]. In the process, we will see explicitly the role of twisting by rational functions.

[1]For reference, see a brief discussion from as recent as February of this year, given in the description for the Clay Mathematics Institute Program "Special Week on Ranks of Elliptic Curves and Random Matrix Theory", available over the internet at www.newton.cam.ac.uk/programmes/RMA/rmaw01.html.

Theorem 1.4. ([RS01], Theorem 3.1). *Suppose that either*

(a) $E[2]$ *has a nontrivial Galois-equivariant automorphism and* $End_{\mathbb{C}}(E) \neq \mathbb{Z}[i]$, *or*

(b) E *has a rational subgroup of odd prime order* p *and* $End_{\mathbb{C}}(E) \not\supset \mathbb{Z}[\sqrt{-p}]$.

Then there is a squarefree polynomial $g(u)$ *of degree* 6 *such that the twist* $E_{g(u)}$ *has rank two over* $\mathbb{Q}(u)$.

This theorem, used in conjunction with the following, will yield results such as our Theorem 1.2. In the following, w is the constant from the functional equation for our elliptic curve's L-series, see page 10 for more.

Theorem 1.5. ([RS01], Theorem 5.1, iii). *Suppose that* E *is an elliptic curve over* $\mathbb{Q}$, *and* $g \in \mathbb{Q}[u]$ *is a nonconstant and squarefree polynomial. Let* $r = \text{rank } E_{g(u)}(\mathbb{Q}(u))$ *and* $k = \lfloor \frac{1}{2}(\deg g + 1) \rfloor$. *Suppose further that the irreducible factors of* g *all have degree at most* 6, *that the Parity Conjecture holds for all twists of* E, *and that there is a rational number* c *such that* $g(c) \neq 0$ *and* $w(E_{g(c)}) = (-1)^{r+1}$. *Then, for* $X \gg 1$, *we have*

$$N_{\geq r+1}(X) \gg X^{1/k}.$$

To see the application of these theorems, we use data from the first theorem in the hypotheses of the second. Assume we have an elliptic curve E over $\mathbb{Q}$ that satisfies either (a) or (b) from Theorem 1.4. Then we can choose $g(u)$ of degree 6 so that $\text{rank } E_{g(u)}(\mathbb{Q}(u)) = 2$. So, in the terminology of the last theorem, we have $r = 2$ and $k = 3$. Throughout this paper, we will allow ourselves to assume the Parity Conjecture (provided we declare when we are doing so), and in particular we assume it for all twists of this E. Then, the only remaining requirement for the result in the preceding theorem to hold is that there exist a c such that $w(E_{g(c)}) = -1$. As we are assuming the Parity Conjecture, this equates to $E_{g(c)}$ having odd rank over $\mathbb{Q}$.

Theorem 1.2, part $(3')$, resulted from a somewhat involved argument which showed that one of the conditions in Theorem 1.4 held and that there would be a number c such that $w(E_{g(c)}) = -1$. Deducing a moral from the above two theorems, though, can already be done without much work. We assume that one of the conditions from Theorem 1.4 holds. Then, assuming the Parity Conjecture, $N_{\geq 3}(X) \gg X^{1/3}$ provided merely that, for some c, $E_{g(c)}$ has odd rank. As we expect to find twists of odd rank frequently (although maybe not as often as half the time for these particular twists, despite what the Density Conjecture suggests in general), this is very believable.

In a sense, the implication of these theorems was a surprise, as many expect $X^{1/4}$ growth. It turns out, though, that almost all of the curves for which this line of argument applies are curves with isogenies. The next two theorems make this more explicit.

Recall the structure of the 2-torsion points on an elliptic curve, $E[2] \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$. Thought of as an $\mathbb{F}_2$-vector space, we can describe the group of automorphisms of $E[2]$. Explicitly, we have $\text{Aut}(E[2]) \cong \text{S}_3$. By definition, an element of $\text{Aut}(E[2])$ is Galois-equivariant if, for every $\sigma \in \text{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$, it commutes with the action of σ on $E[2]$.

The existence of a Galois-equivariant automorphism depends only on the image of the induced map

$$\psi : \text{Gal}(\bar{\mathbb{Q}}/\mathbb{Q}) \to \text{Aut}(E[2]) \cong \text{S}_3.$$

In particular, a non-trivial Galois-equivariant automorphism exists if and only if the image of the above map has a non-trivial centralizer. Thus, Theorem 1.4 applies precisely in the cases when ψ is not surjective. For two of these three cases, our elliptic curve will have an isogeny:

Theorem 1.6. *If the image of ψ has order one or two, then our curve E will have a non-trivial isogeny.*

Proof. If the image has order one, then the entire group $E[2]$ must be fixed by every element in the Galois group $\mathrm{Gal}(\bar{\mathbb{Q}}/\mathbb{Q})$, and so the 2-torsion lies in $\mathbb{Q}$. If the image has order two, then the non-trivial element of the image has to correspond to a two-cycle, not a three-cycle, and so one 2-torsion point must lie in $\mathbb{Q}$.

In either case, we arrive at a rational subgroup of order 2. The completion of the proof follows from the next theorem.

□

The following serves the dual purpose of both completing the last proof, and of showing that the second condition in Theorem 1.4 also implies the existence of isogenies.

Theorem 1.7. *If E has a rational subgroup of prime order p, then E has isogenies.*

Proof. Assume to the contrary that E has no non-trivial isogenies. Let C denote the rational subgroup. Consider the natural projection map $\Phi : E \to E/C$. As complex conjugation fixes C, Remark 4.13.2 in Chapter 3 of [Si86] assures us that Φ is defined over $\mathbb{Q}$. As E has no isogenies (other than to isomorphic curves) we can find a map $\lambda : E/C \to E$ which is an isomorphism. Composing, we get an endomorphism of E, $\lambda \circ \Phi$. But the only endomorphisms for our curve $E(\mathbb{Q})$ are the multiplication by m maps.

The map λ is injective, and so we get on the one hand that the kernel of $\lambda \circ \Phi$ is C, and on the other hand that it is $E[m]$. But the first of these is cyclic (as it has prime order) and the second is not (see p. 89 of [Si86]), and so we have our contradiction. Thus, E must have isogenies.

□

Therein lies the motivation for our interest in curves with no isogenies: It appears as perhaps a coincidence that many of these curves have isogenies, but we are interested in whether the numerical evidence suggests rank 3 frequency among twists of curves with isogenies is similar to the frequency of rank 3 twists among elliptic curves without isogenies.

Now that some general material concerning elliptic curves has been set up, we proceed to the specifics which are used in our calculations directly. We begin in the next section by (briefly) recalling a method for finding rational points on certain quadratic twists of an elliptic curve E, as described in [El94]. He explains the process with reference specifically to the curve $y^2 = x^3 - x$ of conductor 32, but the framework, which we describe in the next section, works just as well in general. Following that, in section 3, we give explanations for both those elements of our method which differ from Elkies', and for certain aspects of Elkies' method which he does not himself elaborate on. Then, in section 4 we describe how the GP/PARI package was used to actually implement this method. Finally, in section 5, we describe some results obtained from using the method to count curves of rank 3 among twists of curves with no isogenies. Along the way, we will have

occasion to use several of the cornerstones of the subject of elliptic curves: the Birch and Swinnerton-Dyer Conjecture (page 11), the Gross-Zagier formula (page 11), Mazur's Theorem concerning torsion groups of elliptic curves (page 22), and the Taniyama-Shimura Conjecture (page 7).

2. Framework of Elkies' Method

We adapt Elkies' construction in [El94] to work for an arbitrary (strong) elliptic curve. For the most part, the details carry over directly, and we may safely refer to the content here as Elkies' method. For those details which don't carry over in as obvious a manner, we still list them here, and save explanations for section 3.

Let E be a strong elliptic curve over $\mathbb{Q}$ of conductor N. To study quadratic twists of E, we will want to look at points in fields of the form $K = \mathbb{Q}(\sqrt{-D})$. For the following construction to work, though, we must put some restrictions on these fields. As the notation suggests, we are interested in *imaginary* quadratic extensions of $\mathbb{Q}$. Furthermore, we require that each prime divisor of the conductor N splits in K. The field K will correspond to the twist E_{-D} in the construction.

To motivate the following, we will need a definition. First, we recall some standard notation. Set the moduli space $X_0(N) := \mathbb{H}/\Gamma_0(N)$ where $\mathbb{H}$ denotes the upper half plane and where

$$\Gamma_0(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL_2(\mathbb{Z}) \mid c \in N\mathbb{Z} \right\}.$$

Consider now a complex lattice $L \subseteq \mathbb{C}$. We can ask for the set $End(L) = \{\alpha \in \mathbb{C} \mid \alpha L \subseteq L\}$. Clearly this set always contains $\mathbb{Z}$, and it is not hard to see that the set is a ring (for instance, that it is closed under multiplication follows from associativity). When this ring is strictly bigger than $\mathbb{Z}$, we will say that the lattice has complex multiplication. We are now ready for the definition:

Definition 2.1. A *Heegner point* is a point τ in the upper half plane $\mathbb{H}$ such that $\mathbb{Z} + \mathbb{Z}\tau$ and $\frac{1}{N}\mathbb{Z} + \mathbb{Z}\tau$ both have complex multiplication, and such that their endomorphism rings are equal.

We think of τ as corresponding to the elliptic curve, group pair $C \cong \mathbb{C}/(\mathbb{Z}+\mathbb{Z}\tau)$, G, where G is the subgroup of C generated by $1/N$. The lattices having complex multiplication corresponds to the elliptic curves C and C/G having complex multiplication, and the rings $End(\mathbb{Z} + \mathbb{Z}\tau)$ and $End(\frac{1}{N}\mathbb{Z} + \mathbb{Z}\tau)$ being equal corresponds to those elliptic curves having the same discriminant.

Our Heegner point definition has us prepared for the following. Let O denote K's ring of integers, and let $I \subset O$ be an ideal (not just a sublattice) such that $O/I \cong \mathbb{Z}/N\mathbb{Z}$. With such an I, we have, for any representative J of an ideal class of O, that $\mathbb{C}/IJ$ and $\mathbb{C}/J$ are both elliptic curves with complex multiplication and with the same discriminant, and that J/IJ is a cyclic subgroup of order N of $\mathbb{C}/IJ$. Working backwards, we will be able to find a Heegner point for each element in the ideal class group. In this manner, we attain h Heegner points, where h equals the class number of K.

Our elliptic curve comes equipped with a modular parametrization $X_0(N) \to E(\mathbb{C})$. (Elkies describes how to ensure this, but thanks to the proof of the Taniyama-Shimura Conjecture, the existence of the map is automatic.) Applying this map to our Heegner points, we get a set of Galois conjugates over our $K = Q(\sqrt{-D})$,

which when summed yields a K-rational point on our elliptic curve E, which in turn yields a $\mathbb{Q}$-rational point on its twist E_{-D}.

The method in general will find interesting points (i.e., non-torsion points). And with our set-up, assuming the Parity Conjecture, a torsion point actually indicates something quite strong: that the quadratic twist E_{-D} has rank at least 3. This will be of interest to us, particularly in section 5, for reasons we described in the previous section.

3. Specifics of our Method

In this section, we elaborate on those elements in section 2 which either differ from the method described in [El94], or which appear there without elaboration. The major differences stem from the fact that he works with twists of a set elliptic curve, while we are interested primarily in other elliptic curves, namely, in the ones with no isogenies.

Our goal here is to pave the way for the the implementation of our method, which we describe in section 4. In particular, this section we describe both how to choose our initial elliptic curve and how to choose appropriate D's to twist by. We describe the parametrization of points of $X_0(N)$ with elliptic curve, group pairs and show that it is well defined. Finally, we use the Gross-Zagier formula to see that, assuming the Birch and Swinnerton-Dyer Conjecture, a torsion point resulting at the end of our process really does imply that the corresponding twist has rank 3.

The first step concerns choosing an elliptic curve E. We restrict ourselves to the strong Weil curve in each isogeny class. (Elkies describes his method, on the other hand, with regards to an elliptic curve which is not the strong curve in its isogeny class. See [Cr97].) In our case, we do it as a simplification. The complication occurs when we consider the modular parametrization $X_0(N) \to \mathbb{C}/L$, where L is the period lattice so that $E(\mathbb{C}) \cong \mathbb{C}/L$. Our map is given by

$$I_\phi(\tau) := 2\pi i \int_{i\infty}^{\tau} \phi(z)dz,$$

where $\phi(z)$, to be described later, is a function depending only on z and on the coefficients for the L-series of our elliptic curve. In particular, $I_\phi(\tau)$ will be the same for each curve in the isogeny class. It is defined as a contour integral which, in general, will be path dependent. So, because isogenous curves can have different complex lattices associated to them, we can only be assured that the map is well-defined if we begin with the strong Weil curve, because that is the one with the densest lattice.

Once we have chosen the initial curve E, our next task is to choose which quadratic twists we would like to consider. Because he was working with the curve $y^2 = x^3 - x$, Elkies did not have to distinguish between a twist by D and a twist by $-D$, as the sign could be controlled by replacing x with $-x$ (and thus the twists would be isomorphic). In general, of course, there will be no such trick. For our method to work, we will need to twist by a negative number, for instance because we will want to consider an elliptic curve $\mathbb{C}/J$ for each J in the ideal class group of $\mathbb{Q}(\sqrt{-D})$, and that only makes sense if J is a complex lattice.

With the restrictions we have already made, the ideal I such that $O/I \cong \mathbb{Z}/N\mathbb{Z}$ can be constructed explicitly. Assume $N = \prod p_i^{k_i}$. We explicitly chose D so that, for each i, the ideal (p_i) splits in $K = \mathbb{Q}\sqrt{-D}$. Write $(p_i) = (\mathfrak{p}_i)(\mathfrak{p}_i')$, where $(\mathfrak{p}_i)$

and $(\mathfrak{p}_i')$ are prime ideals in K. (This is the only possibility in the case where (p) splits: we clearly have $O/pO \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$, and we see that the ideal (p) has index p^2, and so by multiplicativity of indices, if it splits, it splits into two prime ideals.) Then let $I = \prod(\mathfrak{p}_i)^{k_i}$, where the multiplication is multiplication of ideals. By the same sort of index considerations, we have immediately that $O/I \cong \mathbb{Z}/N\mathbb{Z}$.

As J and IJ are ideals in O, the ring of integers of $K = \mathbb{Q}(\sqrt{-D})$, we have trivially that they are both O-modules. An action of O on $\mathbb{C}/J$ and $\mathbb{C}/IJ$ is then well-defined, and in particular our curves have complex multiplication. It is clear that, thought of as lattices, in the spirit of Definition 2.1, we have

$$End(J) = End(IJ) = O \supsetneq \mathbb{Z}.$$

At this point we have the pair $(C, G) = (\mathbb{C}/IJ, J/IJ)$ consisting of an elliptic curve defined over $\mathbb{C}$ and a subgroup G of index N which (we will later see) comes from a Heegner point. Before we can even discuss *finding* the Heegner point, we must first make sure the set-up makes sense.

We define a parametrization of points in $X_0(N)$ to curve, group pairs by

$$\tau \to (\mathbb{C}/(\mathbb{Z} + \mathbb{Z}\tau), 1/N),$$

where we will eventually want to consider a Heegner point τ as an element of $X_0(N)$ instead of as an element of $\mathbb{H}$. Part of the following is needed even to know that such a map is well-defined. There are many variants of the next theorem, see for instance [Ko84] beginning on page 153.

Theorem 3.1. *If τ and $\tau' \in \mathbb{H}$ (not $X_0(N)$) parametrize the pairs (C, G) and (C', G') respectively, then $(C, G) \cong (C', G')$ if and only if there exists $w \in \Gamma_0(N)$ such that $w(\tau) = \tau'$. Here an isomorphism of pairs $(C, G) \to (C', G')$ is defined to be an isomorphism of elliptic curves $C \to C'$ that takes G to G'.*

Proof. $\Rightarrow$ First we assume $(C, G) \cong (C', G')$ and in particular $\mathbb{C}/L_1 \cong \mathbb{C}/L_2$, where here we are making explicit the correlation of an elliptic curve over the complex numbers with the complex plane modulo a lattice, and in particular we are using the notation $C \cong \mathbb{C}/L_1$ and $C' \cong \mathbb{C}/L_2$.

The preceding isomorphism of elliptic curves (ignoring the subgroup) can happen if and only if there exists $\alpha \in \mathbb{C}^*$ such that $\alpha L_1 = L_2$ (see p. 161 of [Si86]) . In our case, we know the lattices explicitly in terms of the accompanying Heegner points, and so it reduces to: $\alpha(\mathbb{Z} + \tau\mathbb{Z}) = \mathbb{Z} + \tau'\mathbb{Z}$. In particular, corresponding to the inclusion $\alpha L_1 \subseteq L_2$ there are integers a, b, c, d such that

$$\alpha\tau = a\tau' + b$$
$$\alpha = c\tau' + d$$

and thus,

$$\tau = \frac{a\tau' + b}{c\tau' + d}.$$

In this manner we obtain a matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ for which $M(\tau') = \tau$. Using the exact same reasoning applied to the reverse inclusion $L_2 \subseteq \alpha L_1$ and solving for τ' and 1 instead of $\alpha\tau$ and α, we get an inverse matrix for M that also has coefficients in $\mathbb{Z}$. We conclude that $M \in SL_2(\mathbb{Z})$.

It remains only to check that $N \mid c$. We haven't yet used the requirement that our isomorphism carry G to G' where both groups are generated by the element $1/N$. This requirement assures that we must have

$$\alpha\frac{1}{N} = (c\tau' + d)\frac{1}{N} = \frac{k}{N} + x + y\tau'$$

where $x, y \in \mathbb{Z}$ and $\gcd(k, N) = 1$. (The last equation comes from the fact that our isomorphism must send a generator to a generator modulo the lattice, but not necessarily to the same generator.) Examining the equation, we see immediately that for τ' to have an integer coefficient, we must have that $N \mid c$, as desired.

$\Leftarrow$ Now we assume that we have a matrix M as defined above, such that $M(\tau') = \tau$. Let $\alpha = c\tau' + d$. Then we have immediately that $\alpha\tau = a\tau' + b \in L_2$ and, even more trivially, that $\alpha \in L_2$. So, $\alpha L_1 \subseteq L_2$.

We can use our knowledge of the determinant of M to get the reverse inclusion.

$$\begin{aligned} d\alpha\tau &= da\tau' + db \\ b\alpha &= bc\tau' + bd \end{aligned}$$

subtracting yields

$$d\alpha\tau - b\alpha = (ad - bc)\tau' = \tau'.$$

Thus $\tau' \in \alpha L_1$. The exact same trick, except multiplying by a and c instead of d and b, also yields that $1 \in \alpha L_1$. Combining the two, we find $L_2 \subseteq \alpha L_1$. Thus, the curves are isomorphic. Does the isomorphism appropriately carry G' to G?

We have $\alpha\frac{1}{N} = c\tau'\frac{1}{N} + d\frac{1}{N} = y\tau' + \frac{d}{N}$. This suffices because $y \in \mathbb{Z}$ as $N \mid c$, and $\gcd(d, N) = 1$ lest d and c share a common divisor, which would force the determinant of M to not equal 1. So, modulo the lattice generated by 1 and τ', we have that $\alpha\frac{1}{N}$ generates the group G', as required. This completes the proof.

□

In section 4, we describe how a Heegner point is actually computed from the knowledge of the elliptic curve, group pair which corresponds to it. (At the current stage of our argument, there is nothing to preclude the possibility that there is no corresponding Heegner point.) Once we have found the Heegner points, though, we can use the modular parametrization to map them to our original curve E. They correspond to a set of Galois conjugates over K in $\mathbb{C}$, and so summing them we attain a rational point of the twist E_{-D}.

In most cases, the point we attain will have infinite order. When instead we attain a torsion point, we have the following:

Theorem 3.2. *Assuming the Birch and Swinnerton-Dyer Conjecture, beginning with a strong Weil curve E of rank 0, the point resulting from the process described in this section will be a torsion point if and only if the corresponding twist E_{-D} has odd rank 3 or greater.*

Proof. Recall the "root number" $w = \pm 1$ from the functional equation for an elliptic curve's L-series. (The Parity Conjecture states that this $w = 1$ if and only if the elliptic curve has even rank.) For a quadratic twist with the condition $\gcd(D, N) = 1$, we have $w_{E_{-D}} = w_E\chi_{-D}(-N)$, where N denotes as usual the conductor of our

elliptic curve E, and where χ_{-D} is the Dirichlet character associated to $\mathbb{Q}(\sqrt{-D})$ (see Section 4.3 in [Si01]).

Part of the subsequent argument requires Class Field Theory, and we must unfortunately resign ourselves to saying that the step whose proof we now omit is "well-known". Our condition that the ideal (p) splits in $\mathbb{Q}(\sqrt{-D})$ for each prime divisor p of N, together with the multiplicativity of characters, assures us that

$$\chi_{-D}(-N) = \chi_{-D}(-1)\prod(\chi_{-D}(p_i))^{k_i} = \chi_{-D}(-1)\prod 1^{k_i} = \chi_{-D}(-1).$$

Calculating $\chi_{-D}(-1)$ requires determining which element complex conjugation corresponds to in the Galois group of $\mathbb{Q}(\sqrt{-D})$ over $\mathbb{Q}$. As $\mathbb{Q}(\sqrt{-D})$ is an imaginary quadratic extension of $\mathbb{Q}$, we see that $\chi_{-D}(-1) = -1$.

Thus, subject to the Parity Conjecture, the ranks of E and E_{-D} have different parities. As E has rank 0, we have in particular that E_{-D} has odd rank.

We can furthermore use the Gross-Zagier formula, as stated in [St88]:

$$\hat{h}(V) = L(E,1)L'(E_{-D},1)/(4\Omega(E)\Omega(E_{-D})).$$

The height $\hat{h}(V)$ is nonzero if and only if we find a non-torsion point. On the right-side, we know $L(E,1) \neq 0$, as the portion of the Birch and Swinnerton-Dyer Conjecture concerning rank 0 curves has already been proven. So, the Gross-Zagier formula tells us that we find a non-torsion point if and only if $L'(E_{-D},1) \neq 0$. Assuming the Birch and Swinnerton-Dyer Conjecture, we have, as desired, that we find a non-torsion point if and only if the rank of E_{-D} is 1.

As we have seen that the rank of E_{-D} is odd, we conclude that a torsion point implies E_{-D} has odd rank greater than or equal to three. (It is this type of argument that leads to estimates of rank frequency which further restrict to ranks of a certain *parity*. See Section 5 in [RS01] for examples.) □

Our task of looking for curves of rank 3 among quadratic twists thus reduces to carrying out the construction and looking for torsion points. In the next section, we will describe a program written using PARI which does precisely that for a large group of appropriate D's, and keeps track of how often a torsion point is found.

The job of identifying a point as a torsion point is simplified in the case which interests us most: namely, when our initial elliptic curve E has no isogenies. However, this is just a pleasant coincidence, and is not the reason for our interest. But we gladly accept the simplification:

Theorem 3.3. *Let E denote an elliptic curve defined over $\mathbb{Q}$, and let E_{-D} denote its quadratic twist over the field $K = \mathbb{Q}(\sqrt{-D})$. If E has no isogenies, then the curve E_{-D} has only one torsion point, the point at infinity.*

Proof. Assume to the contrary that we do have a point $x \in E(K)_{\text{tors}}$ other than the point at infinity. Let n denote the order of x. Consider the subgroup $C = \langle x \rangle$. Let $\bar{x}$ denote the usual complex conjugate of x. As our curve is defined over $\mathbb{Q}$, we can express the multiplication-by-n map in terms of rational functions. As these rational functions send x to the point at infinity, they will also send $\bar{x}$ to the point at infinity, meaning that it is also a torsion point. We can assume that $\bar{x} \in C$, because if it is not, then $x + \bar{x}$ is not the point at infinity, and as $x + \bar{x}$ is a torsion point for which $\overline{x + \bar{x}} = x + \bar{x} \in \langle x + \bar{x} \rangle$, if necessary we can replace x with $x + \bar{x}$.

Now, as we have that $x, \bar{x} \in C = \langle x \rangle$, we note that $\bar{C} = C$, using here as well that our curve is defined over $\mathbb{Q}$. Then complex conjugation fixes the cyclic group C,

and so the exact same argument as was used in the proof of Theorem 1.7 provides us with our contradiction.

□

The following explains this result's importance to us:

Remark 3.4. For an elliptic curve E with no isogenies, to determine if the twist E_{-D} has rank 3 or greater, we need only map the Heegner points to $E(\mathbb{C})$, sum them, and see if the resulting point is on the corresponding lattice, as the curve contains no other torsion points. This, in turn, allows us to make the computations with less precision. (But because our modular parametrization is defined in terms of an infinite series which we only approximate, we cannot expect exact accuracy at this stage. See the discussion in sections 4.8 and 4.10.)

The proof of Theorem 3.3 above also has us in a position to say something about the case when E does have isogenies. Not surprisingly, though, the result is significantly weaker.

Theorem 3.5. *If there is n-torsion in $E_d(\mathbb{Q})$, then E has an isogeny with cyclic kernel of order n.*

Proof. Let x be a torsion point of order n. Consider x as a point of $E(K)$. We divide into two cases:

(1) $\bar{x} \in \langle x \rangle$. In this case, $\langle x \rangle$ is defined over $\mathbb{Q}$. Then we are in the same situation as in Theorem 3.3 and Theorem 1.7 before it. In these, we saw how to attain $\langle x \rangle$ as the kernel of an isogeny, and it clearly is cyclic of order n.

(2) $\bar{x} \notin \langle x \rangle$. We know that $\bar{x}$ has order n. These two facts are already enough to assure us that x and $\bar{x}$ generate $E[n] \cong \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. We now consider, as in Theorem 3.3 above, $\langle x + \bar{x} \rangle$, only this time we pay attention to its order. As $x + \bar{x}$ corresponds to the point $(1, 1) \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, we see that its order is n. Thus, we have reduced to the same case as in (1).

□

With this set-up, it is not hard to begin making estimates for the function $N_{\geq 3, odd}(E)$. Because we are checking only a subset of the possible twists, though, we must decide whether or not to expect the same results with our current set-up as we would get if we had a way to check *all* the twists in a given range.

In a sense, the easy answer is "yes". We don't know of any numerical evidence to suggest that among twists of *odd* rank (the importance of the twists' ranks being odd will become clear below), the higher ranks will appear in our particular twists any more or less frequently than they appear in the rest. The twists we check do have odd rank, though (see Theorem 3.2), and curves of rank 3 or higher *do* appear more frequently among odd twists than they do overall. So, we can find the proportion of curves we check which have rank 3 or higher, and roughly half of that should be the proportion of all imaginary twists having odd rank three of higher (subject to the Density Conjecture). Then again, since we just want to compare against results of the form $N_{\geq 3}(X) \gg X^a$, a factor $1/2$ won't make any difference.

In another sense, though, often we can do better . Referring back to the proof of Theorem 3.2, we find that, subject to the Parity Conjecture, twists by $-D$ with *odd* rank are precisely those for which $\chi_{-D}(N) = 1$. We are checking only a subset

of those, the ones for which $\chi_{-D}(p) = 1$ for each prime divisor p of N. But, at least when our conductor is prime, there is not much left to be done. The only cases we are avoiding are twists by numbers which are not relatively prime to that conductor.

We can resolve this by repeating the process with the twist E_{-N}, provided it has rank zero. That twist has conductor N^2, and so we won't be losing any twists by appropriate D's (those which are divisible by N and not divisible by N^2). As a twist by a square is isomorphic to the original elliptic curve, we don't have to repeat the process. Thus, at least in the case where E has a prime conductor, it's not hard to extend the process to include all possible twists in whatever range interests us.

4. Implementation

In this section, we describe how the methods of sections 2 and 3 were actually carried out using the GP/PARI package. Our general philosophy for dividing material between this section and the last has been to include in section 3 those results which are needed to ensure the method will work but which don't actually factor in to any calculations, and to include in this section those results which are used in the program itself. In fact, there is a third division. Occasionally, for the program to work correctly certain cases which from a mathematical point of view are treated exactly the same must be dealt with separately. For instance, if the field $K = \mathbb{Q}(\sqrt{-D})$ has trivial ideal class group, then the output of PARI's function which finds K's class group structure is different from the output in the other cases. So, the two cases need to be accounted for separately. Details such as this, which stem from the specific functions in PARI we use, do not seem of interest mathematically, and such material is omitted from both this section and the last. To compensate, the code of the final program, complete with comments at such junctures as described above, is included at the end of the paper.

The method we have described thus far shows how, for certain twists of a strong elliptic curve of rank 0, to determine which ones have rank 3 or higher with a very high probability of success. Using the PARI program, the method can be repeated consecutively for many D's, and in practice this is what we do. This is the sensible method, as each D with the current program takes no more than a few seconds for elliptic curves of conductor around 100, and D's up to about 100000.

Our program divides naturally into pieces:

4.1. **Elliptic Curve Calculations.**

We deal with both a fixed elliptic curve, and with many twists of that curve. In general, almost every aspect of the program will need to be repeated for each twist. However, there are certain calculations which only need to be done once because they concern only the initial elliptic curve E, and we list those here.

- Find and factor N, the conductor of our elliptic curve. PARI has commands for both of these.
- Create a list with a sensible number of coefficients in the L-series of E. Again, PARI can do this for us. The precise number of coefficients we need will depend on the twist, but we can always find more coefficients if necessary. In practice, for our ranges, 10000 coefficients has been more than enough, and these can be computed almost instantaneously.

- Use PARI to find the generators of the complex lattice associated to our elliptic curve.
- Set the range of D's representing the quadratic twists that we want to check through.

4.2. **Finding Appropriate D's.** As stated in section 2, for D's in our range, we are only interested in those D's for which, given any prime divisor p of the conductor N, the ideal (p) splits in the field $K = \mathbb{Q}(\sqrt{-D})$. We also, as described above, require that D be squarefree and that the field K be an imaginary quadratic field. Finally, we ignore as special cases those where $-D = -1$ or -3. These have the potential to complicate matters because their corresponding fields contain extra roots of unity. Furthermore, our eventual goal is to understand the rank distributions among thousands of D's, and so being able to include a few extras is not worth the trouble. Back to the first requirement, which leads to the question: In which fields does a prime ideal split?

Proposition 4.1. *The ideal (p), for odd prime p, splits in the ring of integers O of the imaginary quadratic field $K = \mathbb{Q}(\sqrt{-D})$ if $-D$ is not divisible by p and is a quadratic residue modulo p. Furthermore, if a is a square-root of $-D$ modulo p, then the ideal $(a + \sqrt{-D}, p)$ is one of the two prime ideals lying above (p).*

Proof. We assume $-D$ is a quadratic residue modulo p. Choose a so that $a^2 \equiv -D \mod p$. As usual, let K denote $\mathbb{Q}(\sqrt{-D})$ and let O denote K's ring of integers. O has two generators, and as additive groups $O/pO \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$. So, the ideal (p) has index p^2 in O.

We clearly have

$$O \supseteq (a + \sqrt{-D}, p) \supseteq (p).$$

Considering indices, which are $1, ?, p^2$ in O respectively, using the fact that the indices must satisfy the usual division properties, we see that either we have equality on one side, or $(a + \sqrt{-D}, p)$ is a prime ideal lying above (p).

First we'll show that $(a + \sqrt{-D}, p) \neq (p)$. Assume to the contrary that, in particular, $a + \sqrt{-D} \in (p)$. As (p) is a principal ideal, this can occur only if p divides both a and $\sqrt{-D}$. The latter provides us with a contradiction. (This relies on the fact that we've forced p to be an *odd* prime. If $p = 2$ and O contains $(1 + \sqrt{-D})/2$, which happens when $-D \equiv 1 \mod 4$, this is no longer true.)

Now we'll show that $(a+\sqrt{-D}, p) \neq (1)$. Recalling the notation on page 9, denote $(a+\sqrt{-D}, p)$ by $\mathfrak{p}$. We claim that $(p) \supseteq (a-\sqrt{-D})\mathfrak{p}$. To prove this, it is enough to observe that $(a-\sqrt{-D})p \in (p)$ and $(a-\sqrt{-D})(a+\sqrt{-D}) \in (p)$. The first of these is automatic. To see the second, we expand $(a - \sqrt{-D})(a + \sqrt{-D}) = a^2 + D \equiv 0 \mod p$ by how we chose a. Thus, we have as desired that $(a - \sqrt{-D})\mathfrak{p} \in (p)$. We are now ready to prove that $\mathfrak{p} = (a+\sqrt{-D}, p) \neq (1)$. If to the contrary we did have equality, then we could use our above equation to find that $(a - \sqrt{-D}) * 1 \in (p)$ which we have already seen does not happen (provided again that p is an *odd* prime).

□

We now have a good criterion which we will use for choosing our D's, namely, we will choose squarefree D's such that D and N are relatively prime, and such that $-D$ is a square modulo each odd prime divisor of N. Note that, although there may be a faster way to determine this than by the brute force method of actually

finding the square-roots, since we will eventually want the ideals lying above (p) explicitly, we might as well find the square roots a as we go.

In the case N is even, we will see in the next section that the requirement (2) splits will be equivalent to the requirement that $-D \mod 8 \equiv 1$. Assuming this, we now know how to choose our D's.

4.3. **Creating the Ideal of Index N.** As described in section 3, the ideal I of index N which we need on the way to finding our Heegner points can be computed easily (by PARI) once we know the prime ideals lying above (p) for each prime divisor of N. Above we described how to find such ideals for the case of an odd prime p. Consider now the prime ideal (2).

For squarefree $-D$, the ideal (2) ramifies in the cases when $-D \not\equiv 1 \mod 4$ (this follows for example from Theorem 4.8.8 in [Co93] and the fact that the discriminant is even in the case $-D \not\equiv 1 \mod 4$.) Furthermore, the case $-D \equiv 1 \mod 4$ divides into the cases $-D \equiv 1 \mod 8$ and $-D \equiv 5 \mod 8$, which must be handled differently. In the first case, (2) splits, but in the second case, (2) is already prime (these follow from Corollary 4.8.12 in [Co93] and the fact that $(\frac{-D}{2})$ equals -1 or 1 according as $-D$ is congruent to 1 or 5 mod 8 respectively, as on p. 28 of [Co93]). So, we only consider the case for which $-D \equiv 1 \mod 8$.

From here, it is easy to use commands in PARI to both find an ideal lying above (2), and to subsequently multiply our ideals together to obtain the ideal I. (This is an example of where small details, like using PARI to find an ideal lying above (2), have been omitted, and where the interested reader can look directly to the program at the end.)

4.4. **Finding Class Group Representatives.** We would now like to find a representative J of each element in the ideal class group. PARI has a function which finds generators for the ideal class groups as binary quadratic forms. We can then use the following formula from p. 221 of [Co93] to convert the quadratic form into an ideal: We associate to a quadratic form (a, b, c) the ideal $a\mathbb{Z} + \frac{-b+\sqrt{D}}{2a}\mathbb{Z}$, where D is the discriminant $b^2 - 4ac$ of the quadratic form. Then p. 222 of [Co93] assures us that the above association is a surjective homomorphism from the class of binary quadratic forms to the group of ideal classes. The only subtlety (if it may be called that) for implementing this step is that we must represent the ideal in terms of a $\mathbb{Z}$-basis, which requires treating the case $-D \equiv 1 \mod 4$ separately, as the basis for its ring of integers is different. (Again, the actual program code should be consulted if details concerning such matters are desired.)

Once we have our list of quadratic forms which generate the ideal class group, we still need to turn that data into a list of all the elements in the class group. Although trivial, I found this required more thought than I would have expected. PARI provides us with a set of generators $(g_1, \ldots, g_k)$, together with the corresponding finite abelian group structure $(m_1, \ldots, m_k)$ (which corresponds to $\mathbb{Z}/m_1\mathbb{Z} \oplus \ldots \oplus \mathbb{Z}/m_k\mathbb{Z}$), where k is the number of generators. If these m_j's were relatively prime, we could use the Chinese Remainder Theorem to justify an easy method for finding all the elements: start with a vector of size h, where h is the class number of O, and put in the ith position the element $g_1^{i \mod m_1} \cdots g_k^{i \mod m_k}$. But in general we won't have relatively prime m_j's, and the solution won't be as clean. Not surprisingly, though, "mod"'s will still appear in the exponents.

The idea is as follows:

- Begin with a vector of length n.
- Divide the vector into m_1 blocks of length h/m_1.
- Place g_1 at each position in the first block, g_1^2 at each element in the second block, and so on until the m_1st (and last) block, in which we place $g_1^{m_1 \mod m_1} = g_1^0$ at each position.
- Divide the resulting vector into $m_1 m_2$ blocks of length $h/m_1 m_2$.
- Multiply each element in the first block by g_2, and generally, each element in the jth block by $g_2^{j \mod m_2}$.
- Repeat this process in the obvious way for each summand in the given class group structure.
- Noting that $\prod_{i=1}^{k} m_i = h$, it is clear after a little thought that this method will produce a vector with a different element of the ideal class group in each of its components.

To maximize efficiency in the program, we found it best to work with quadratic forms through this stage, and only then convert to ideals in the way described above. This enables us to reduce the quadratic forms (using PARI) as we go, which proves to be absolutely critical in the case when the ideal class group has large cyclic components (around 400 terms, for instance). In section 4.7, we will describe a method for altering the Heegner points we eventually find. The process comes down to acting on the Heegner points with elements of $\Gamma_0(N)$ in such a way that increases their imaginary components. In practice, we find that reducing the quadratic forms at this stage saves much of the work that would have to be done later. Again, see the code at the end for details of how we actually use PARI to reduce the quadratic forms.

4.5. **Finding IJ ideals.** This is one of the shortest steps. For the ideal I found in 4.3 and for each ideal J found in 4.4, we want to find the sublattice IJ. As PARI can perform ideal multiplication, this can be done immediately.

4.6. **Working Backwards to Find the Heegner Points.** At this point, we have lattices $J \supseteq IJ$ such that $J/IJ \cong \mathbb{Z}/N\mathbb{Z}$ (these were proven in section 3). What we are interested in is the corresponding Heegner point: the point τ such that $\mathbb{C}/IJ \cong \mathbb{C}/(\mathbb{Z}+\tau\mathbb{Z})$ and so that $\mathbb{C}/J \cong \mathbb{C}/(\frac{1}{N}\mathbb{Z}+\tau\mathbb{Z})$. We now describe the process for finding such a τ.

As we already noted (see p. 9), the above isomorphisms of elliptic curves correspond to the following equivalences of lattices: $IJ = \alpha(\mathbb{Z}+\tau\mathbb{Z})$ and $J = \beta(\frac{1}{N}\mathbb{Z}+\tau\mathbb{Z})$ for some non-zero α and β in $\mathbb{C}$. Our isomorphisms defined by α and β should clearly take $\mathbb{Z}+\tau\mathbb{Z}$ to the same place (viewing this as the lattice α acts on, and a *sublattice* that β acts on). Just the fact that the element 1 is mapped to the same place in both cases is enough to already assure us $\alpha = \beta$.

PARI provides us with a $\mathbb{Z}$-basis for our lattices J and IJ. Say we have

$$w_1\mathbb{Z} + w_2\mathbb{Z} = J$$
$$u_1\mathbb{Z} + u_2\mathbb{Z} = IJ$$

and further,

$$r_1 w_1 + r_2 w_2 = u_1$$
$$s_1 w_1 + s_2 w_2 = u_2.$$

Clearly we can't have both w_1 and $w_2 \in IJ$, so we can safely assume $w_1 \notin IJ$.

We need a lemma:

Lemma 4.2. *If we have $IJ \subseteq J$ with generators as given above, then the index of IJ in J is equal to the determinant of the matrix* $\begin{pmatrix} r_1 & r_2 \\ s_1 & s_2 \end{pmatrix}$.

Proof. The result is well-known, and we omit its proof, which follows similar lines to the argument given below, where we choose clever generators of the lattices to simplify things. □

The following, an immediate consequence of our lemma, will be used in the subsequent argument.

Remark 4.3. As $J/IJ \cong \mathbb{Z}/N\mathbb{Z}$, we have that $\det \begin{pmatrix} r_1 & r_2 \\ s_1 & s_2 \end{pmatrix} = N$.

The basic idea behind our construction of τ is that we would like one of J's two generators to already be in IJ. Then that generator will play the role of $\alpha\tau$ while J's other generator can play the role of $\frac{\alpha}{N}$. Once we have such a set-up, we can immediately solve for τ. Of course, for the generators PARI gives us, there is no reason to expect such a property will already be satisfied.

Let n denote the product of all primes p dividing N and not dividing $\gcd(r_2, s_2)$. We will replace w_1, the generator of J, with $w_1 + nw_2$. (Clearly as a $\mathbb{Z}$-basis, $(w_1 + nw_2, w_2)$ generates the same lattice as (w_1, w_2).) We now have the new formulas,

$$r_1(w_1 + nw_2) + (r_2 - nr_1)w_2 = u_1 \quad (1)$$

$$s_1(w_1 + nw_2) + (s_2 - ns_1)w_2 = u_2. \quad (2)$$

Lemma 4.4. *We claim that $(r_2 - nr_1)$ and $(s_2 - ns_1)$ are relatively prime.*

Proof. Equations 1 and 2 above, just as we described earlier with reference to the original basis, define a matrix whose determinant is N. For this reason, the only possible divisors of $(r_2 - nr_1)$ and $(s_2 - ns_1)$ are divisors of N. Consider first a prime divisor p such that p does not divide $\gcd(r_2, s_2)$. Say p does not divide r_2. We chose n so that p does divide n in this case, so p does not divide $(r_2 - nr_1)$.

The other case is only slightly more difficult. Assume we have a prime p which divides N such that p does divide $\gcd(r_2, s_2)$. By construction, p will not divide n, but if it divides both r_1 and s_1, then it will divide $(r_2 - nr_1)$ and $(s_2 - ns_1)$.

So, we will assume that p divides each of r_1, r_2, s_1, s_2 and arrive at a contradiction. Recalling the original appearance of these coefficients, we see that such a property would imply

$$(u_1, u_2) \subseteq (pw_1, pw_2) \subseteq (w_1, w_2).$$

But such inclusions (of abelian groups) would lead to the following subgroup:

$$(w_1, w_2)/(pw_1, pw_2) \leq (w_1, w_2)/(u_1, u_2) \cong \mathbb{Z}/N\mathbb{Z}.$$

But $(w_1, w_2)/(pw_1, pw_2) \cong \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$, and we know that $\mathbb{Z}/N\mathbb{Z}$ has only cyclic subgroups, which provides us with our contradiction. Thus, $(r_2 - nr_1)$ and $(s_2 - ns_1)$ are relatively prime.

□

So, we have equations

$$\begin{aligned} r_1(w_1 + nw_2) + (r_2 - nr_1)w_2 &= u_1 \\ s_1(w_1 + nw_2) + (s_2 - ns_1)w_2 &= u_2 \end{aligned}$$

with the w_2 coefficients relatively prime. Thus, we can use PARI, which uses the Euclidean algorithm, to find integers c and d such that

$$cr_1(w_1 + nw_2) + ds_1(w_1 + nw_2) + w_2 = cu_1 + du_2.$$

We can now change generators once again, this time to $(w_1 + nw_2)\mathbb{Z} + (cu_1 + du_2)\mathbb{Z} = J$. As $(cu_1 + du_2) \in IJ$, and we know $J/IJ \cong \mathbb{Z}/N\mathbb{Z}$, we have that $N(w_1 + nw_2) \in IJ$. Consider the two lattices $(N(w_1 + nw_2), cu_1 + du_2)$ and IJ. We have that $(N(w_1 + nw_2), cu_1 + du_2) \subseteq IJ$ and both have index N in $((w_1 + nw_2), cu_1 + du_2) = J$, and so they are equal. Summarizing, we have:

$$\begin{aligned} (w_1 + nw_2)\mathbb{Z} + (cu_1 + du_2)\mathbb{Z} &= J \\ N(w_1 + nw_2)\mathbb{Z} + (cu_1 + du_2)\mathbb{Z} &= IJ. \end{aligned}$$

We have already described where to go from here. We attain immediately that $\alpha = N(w_1 + nw_2)$ and then $\tau = (cu_1 + du_2)/\alpha$ is almost our desired Heegner point. The only catch is that we have not yet made sure it indeed lies in the upper-half plane. This is easily remedied, though. First, note that we cannot have $\tau \in \mathbb{R}$, lest the two generators of IJ both lie on the same line. The other possibility is that τ lies in the lower-half plane, in which case we can replace τ with $-\tau$, because they clearly will both determine the same lattice $\mathbb{Z} + \tau\mathbb{Z}$.

4.7. **Making the Heegner Points More Manageable.** We now have our list of Heegner points, which in section 4.8 we will map to our original elliptic curve E using the modular parametrization. The process in 4.8 takes more time the smaller the imaginary part of the Heegner point. We think of our Heegner points as elements of $X_0(N)$, though, and so we can act on any Heegner point with any element of $\Gamma_0(N)$ in an attempt to increase its imaginary part. We describe the process in this section.

Say $\tau' = \frac{a\tau+b}{c\tau+d}$. We would like to choose a, b, c and d so that τ' has a larger imaginary part than τ. Page 269 of [Ah79] gives that

$$\text{Im } \tau' = \frac{\text{Im } \tau}{|c\tau + d|^2}.$$

Our desire to increase τ's imaginary part thus corresponds to finding integers c, d such that:

(1) $|c\tau + d| < 1$,
(2) c is divisible by N,
(3) c and d are relatively prime.

In certain cases, PARI can be used to make the process very simple. This comes from the observation that finding c and d so that $|c\tau + d|$ is small and c divisible by N corresponds to finding a short vector in the lattice generated by $(N\tau, 1)$. In terms of *this* basis, the coefficients c' and d' will always be relatively prime (because if they weren't, one could divide by their gcd to find a smaller vector). The problem

is that Nc' and d' might have common factors. As a simple example, this happens about half the time when N is a power of 2.

PARI has a method to find a short vector in a lattice (see the LLL algorithm described in [Co93] for a precise statement), and so for each of our Heegner points, we first check to see if PARI can immediately solve our problem. When it can't, we need to do things by hand.

In practice, the following method has worked satisfactorily; however, this remains a time-consuming part of the program. Also, there are certain steps along the way that could be altered without any clear effect on the program (for instance, we will be repeating our method numerous times, and the question arises as to how often to repeat the initial PARI lattice check, described above). When limited experimentation has not made it clear which path is better, the decisions have been somewhat arbitrary.

For a chosen c divisible by N, it is not hard to choose the d which minimizes $|c\tau + d|$. (Occasionally two d's may be practical, but our program only finds the better of the two.) When the imaginary part of τ is tiny, the case we need to be most concerned with, we will almost always be able to get $|c\tau + d| < 1/2$, which will correspond to multiplying the imaginary part of τ by four, which is quite good.

We proceed, then, in the cases when PARI finds lattice coordinates which are not relatively prime, by choosing our coefficients $c'N$ incrementally through $c' = 1, 2, \ldots$, and stopping when we find one for which $|c\tau+d| < 1$ with c and d relatively prime. In particular, we make no attempt to optimize the c, d pair, as multiplication by four already seems quite good. When we find a good pair, we replace τ by the corresponding τ' (we can use PARI's Euclidean algorithm function to find the a and b coordinates, because we know the determinant of our matrix must be 1) and then we repeat the process for that τ'.

A subtlety comes in knowing when to stop: both when to stop looking for $c'N$ with a good corresponding d, and when to stop repeating the process. Of course, when we stop looking for an appropriate $c'N$, we have no choice but to stop repeating the process (because the repetition would begin by looking for a $c'N$!) As to when we should stop looking for a good $c'N$, that also has a precise and easy to understand answer. Once $c'N * \mathrm{Im}(\text{Heegner}) > 1$, there is no need to look at larger $c'N$'s.

The last question is when we should stop even if we do keep finding appropriate $c'N$'s. In practice, we find that a Heegner point with an imaginary part of around .01 can be mapped to a curve point very easily. So, assuming that each success really does multiply our imaginary part by 4, as explained above, we can find how many times the process would have to be repeated in order for the imaginary part to grow beyond .01. The assumption that it will be multiplied by four will be below the expected growth rate at the beginning, but above towards the end, as the imaginary part of τ becomes more and more of a factor in the $c\tau + d$ term. In practice, though, we find that this does an efficient job of both suitably raising the imaginary part of the Heegner point, and of doing it in a suitable amount of time.

4.8. **Finding the Corresponding Curve Points.** We now have our Heegner points, and in this section we will describe how to find the corresponding points on $E(\mathbb{C}) \cong \mathbb{C}/L$.

Following [El94], let $q = \exp 2\pi i\tau$ and let

$$\phi(\tau) := \sum_{n=1}^{\infty} a_n q^n,$$

where a_n is the nth coefficient of E's L-series. Elkies defines the map

$$I_\phi(\tau) := 2\pi i \int_{i\infty}^{\tau} \phi(z)dz,$$

defined on the upper half plane, which descends to our modular parametrization. (Elkies does this for any *modular* elliptic curve over $\mathbb{Q}$, which thanks to the proof of the Taniyama-Shimura Conjecture, is no longer a restriction.) Merely combining the two formulas yields:

$$I_\phi(\tau) = 2\pi i \int_{i\infty}^{\tau} \sum_{n=1}^{\infty} a_n q^n dz,$$

where now $q = \exp 2\pi iz$. Noting that we can take our range of integration to lie entirely in the upper-half plane, and that the coefficients a_n of the L-Series grow like $\sqrt{n}$ (see [Si94], I.11.2.1), we see that the series converges absolutely. So, we are justified in exchanging the order of summation and integration, which yields:

$$\begin{aligned} I_\phi(\tau) &= 2\pi i \sum_{n=1}^{\infty} a_n \int_{i\infty}^{\tau} q^n dz, \\ &= 2\pi i \sum_{n=1}^{\infty} a_n \frac{\exp(2\pi izn)}{2\pi in}\Big|_{z=i\infty}^{\tau} \\ &= \sum_{n=1}^{\infty} a_n \frac{\exp(2\pi n\tau) - \exp(-2\pi n\infty)}{n} \end{aligned}$$

So, we attain, as in [El94], the equation

$$I_\phi(\tau) = \sum_{n=1}^{\infty} \frac{a_n}{n} q^n.$$

We would like to have a way of calculating the images of our Heegner points to various degrees of precision depending on what we would like to do with the data. If we just want to know if it's near a lattice point, then we don't have to have as much precision as if we actually want to have a good chance of calculating the values of the corresponding curve points (x, y) in the field K.

We will describe how to obtain an estimate for the number of terms needed in the sum. Say we would like n decimal places of accuracy. Then, we can try cutting off the sum after $|a_y \frac{\exp 2\pi iy\tau}{y}| < 10^{-n}$, where we hope that sufficient cancellation among the later terms will keep this a respectable estimate. This also doesn't take into account the number of times we will be repeating the calculation before summing them and finding our point of interest. Here also, though, we can hope that we will be overestimating as often as we are underestimating, and so it hopefully won't be *too* bad of an estimate. To be safe, in practice we will ask for several more digits of accuracy than we actually expect.

Even with these compromises, though, the size $|a_y \frac{\exp 2\pi iy\tau}{y}|$ seems difficult to gauge. But we have still that the L-series coefficients grow like $\sqrt{y}$, and so the formula becomes much more manageable, where again we hope our precaution of asking for extra accuracy will make up for imprecision in the $a_y \sim \sqrt{y}$ estimate.

Since the modulus of $\exp(2\pi iy\tau)$ is all that concerns us here, we can restrict to worrying about $\exp(-2\pi y \mathrm{Im}\tau)$. From here, we can easily use PARI to find a good estimate on the number of terms in the sum we should take. Furthermore, it is now apparent why we took the trouble in section 4.7 of increasing the imaginary part of τ.

4.9. **Sum the Points.** Now that we have class number different points of $E(\mathbb{C})$, we want to sum them so that we can find a number defined over K. We have points in $\mathbb{C}/L$, and so summing them is very quick.

4.10. **Check for Triviality.** This step varies depending on the initial curve E. In the case where E has no isogenies, as we saw in section 3, we need only check to see if the resulting point is a lattice point (i.e., corresponds to the point at infinity on the curve.) It's easy to find the coordinates of our resulting point in terms of $E(\mathbb{C})$'s associated lattice: Our curve is defined over $\mathbb{Q}$, and thus the lattice is fixed under complex conjugation (this follows from consideration of the function which produces the lattice) and so we can take one of the lattice generators to be real. Finding lattice coordinates is then easy because the imaginary component of our point will immediately give us one of the two coordinates.

We can't expect exact precision, though, because our points were obtained via a finite number of terms in an infinite sum. We adopt the strategy of checking to see if the coordinates are "close enough" to integer coordinates, and resign ourselves to the fact that occasionally a point may lie in this region even if it is not a lattice point. Since our goal is to judge a proportion of curves of rank 3, this is not a big concession. If we are searching through x twists looking for the ones with rank 3 or higher, then we look for points which are less than about $1/x$ units away from a lattice point. Then, assuming that either the real or the imaginary part of a non-torsion point is randomly distributed, we would expect no more than about one false torsion point over the whole range, and probably not even that, as we aren't considering the twists with even rank. In practice, assuming that either the real or the imaginary part is randomly distributed has appeared to be an accurate assumption, while assuming that the real and imaginary parts were randomly distributed independently of each other has appeared to be a false assumption (see Remark 5.2).

If E does have isogenies, then we must be more careful. As described towards the end of section 3, for a particular E, we do not have too many different types of torsion to worry about. In particular, we will have n-torsion only in the case that E has an isogeny with cyclic kernel of order n. Even if we do have torsion, though, often this is not much to worry about.

Theorem 4.5. *Assume E has an isogeny with cyclic kernel of prime order p but does not have one with order p^2. In this case, rational p-torsion can appear in at most three different twists.*

Proof. Let x and y denote p-torsion points in different fields $\mathbb{Q}(\sqrt{-D})$ and $\mathbb{Q}(\sqrt{-D'})$. Then they are linearly independent (as points of $E(\bar{\mathbb{Q}})$). From our knowledge of the

group $E[p]$, we can deduce that these will generate all the p-torsion points. Considering both x and y as elements of $E(\mathbb{Q}(\sqrt{-D}, \sqrt{-D'}))$, we know that their linear combinations will lie in some (not necessarily proper) subfield of $\mathbb{Q}(\sqrt{-D}, \sqrt{-D'})$. But there are only three quadratic extensions of $\mathbb{Q}$ in this range.

□

Complications may arise when we are dealing with torsion points with order a power of a prime. If need be, we can always multiply our point by some m if we are concerned about finding m-torsion points. The drawback to this is that it requires that our calculations be done with more precision. Even with such considerations, though, as Mazur's Theorem puts such a strict bound on the size of the torsion subgroup, we can temper how much we are concerned with such matters.

4.11. **Summarize and Repeat.** We are interested in keeping track of how frequently we find trivial points, and so we should keep some sort of variables which remember how many D's have been checked and how many trivial points have been found. We add 1 to the variable remembering the number of D's checked, and if we found a trivial point above, then we add 1 to the other variable. For purposes of having flexible records, each time we find a trivial point, we write the corresponding D, the number of D's checked, and the number of trivial points found to a file. Such a file will be useful for analyzing the data.

Assuming we are still within the range of D's set at the very beginning of the process, we now go through all but the initial step again.

5. Results

To a certain extent, testing our program is not difficult. The point we eventually find should correspond to a point of $E(K)$, where $K = \mathbb{Q}(\sqrt{-D})$. So, once we have our final point of $\mathbb{C}/L$, we can use Pari (which uses the Weierstrass $\mathfrak{p}$-function) to find that corresponding point. If our curve has small conductor (less than 50, say) and $D < 30$ or so, it is not hard to recognize the coordinates of the resulting point as elements of K, assuming nothing has gone wrong. Even with larger values of N and D, as long as they don't exceed 100 or so, we can easily use Pari's continued fraction function to see if the real parts of our coordinates seem to be rational. (Dividing by $\sqrt{D}$, we can do the same thing for the imaginary parts.) This is the method which we first used to test our program.

Of course, simply finding points that lie in K does not guarantee that our program is doing what it should. There is a better testing method, but it is unfortunately limited to the specific curve $E_1 : \{y^2 = x^3 + 4x\}$, which is the strong Weil curve in the isogeny class of $E_2 : \{y^2 = x^3 - x\}$, the conductor 32 curve Elkies studied. He lists[2] all of the twists by D's less than 10^6 which his method suggests have odd rank 3 or higher. For D's less than 65000, our program identifies precisely the same curves as his. We did not test any larger D's with this curve.

Remark 5.1. Our comparison relies on the fact that twists of isogenous curves by the same D are isogenous, and thus have the same rank. To see this, consider the isomorphisms $\Phi : E_d \to E$ and $\Psi : E'_d \to E'$, both defined over $\mathbb{Q}(\sqrt{-D})$. If $\lambda : E \to E'$ is an isogeny (defined over $\mathbb{Q}$, as usual), then we consider the map $\Psi^{-1} \circ \lambda \circ \Phi : E_d \to E'_d$. We would like to say that this map is defined over

[2]See http://www.math.harvard.edu/~elkies/cong_r3_7a.html.

$\mathbb{Q}$. It is clearly defined over $\mathbb{Q}(\sqrt{-D})$. We described the isomorphisms Φ and Ψ explicitly all the way back on page 1, and the description there makes clear that the composition of Φ with complex conjugation equals $-\Phi$ (from the group law). Similar reasoning applies to Ψ. Thus, considering this for both Φ and Ψ^{-1} and observing the cancellation, we get that the above map really is defined over $\mathbb{Q}$, and so the twists really are isogenous.

5.1. **Alterations to our Described Method.** The process described in section 4 is not always optimal. To attain the results we list below, we altered the process in two ways. The spirit of the method, though, lies completely in the process described in section 4, and we include this subsection only for the sake of completeness. Of our two changes to the method, one concerns the efficiency of the program, and one is a rather *ad hoc* solution to a problem arising in rare cases.

How we increase the efficiency of the program is simple. As the slowest part of the program occurs when we apply the modular parametrization to the Heegner points, we simply lessen the amount of accuracy we ask for in this stage. We then attain a list of potential rank 3 twists, which we repeat the process with but to a greater degree of precision. For the results given below, we looked for points having coordinates within .005 of a lattice point on the first run, and used four digits of precision that time, and within 5×10^{-10} of a lattice point on the second run, and used 12 digits of precision then. (Here, when we speak of digits of precision, we mean in the sense of section 4.8.) Testing against Elkies curve, we added another decimal place of precision in the second run, to make up for the fact that his curve has 2-torsion and we thus had to double the resulting points.

Remark 5.2. Our evidence, though severely limited, suggests that the coordinates of our complex point, if random, are nevertheless not independent of each other. For instance, with the conductor 115 curve described below, the original .005 bounds produced one "false" rank 3 twist for about every 200 D's checked. If we assume only one of the coordinates is random, we would expect an error about every 100 times (the interval doubles to .01 when we consider that the error can occur to either side of the lattice point). If both were random and independent, we would expect an error about every 10000 times.

What we now describe is less reasonable. On occasion, the program described above and in the code at the end of this paper finds a Heegner point for which, to map using the modular parametrization with any accuracy would take millions of terms, whereas it normally would not take more than 7500 or so. (This happens rarely, although the twists for which it happens seem to cluster together.) The problem occurs on the rare occasions when the real part of the Heegner point is significantly smaller than the imaginary part. In such instances, when our program chooses the d that minimizes $|c\tau + d|$, it chooses 0 for every c less than a certain bound, which clearly prevents $\gcd(c, d) = 1$. By the time c is big enough that $d \neq 0$, the imaginary part of $c\tau + d$ will already be greater than one, at which point our program terminates.

In practice, this has not been hard to fix, although the method is hard to justify. In every case encountered thus far, we have been able to solve the problem by eliminating the quadratic form reduction from the process, and proceeding immediately to the process where we try to increase the imaginary component by hand (as was described in section 4.7, and is currently used only after we reduce the quadratic

forms). This has worked, perhaps because the latter method makes no effort to find a small real part (I do not know if that is done implicitly somewhere in the former method), or perhaps simply because the probability of two different methods both producing a tiny real part is significantly smaller than the probability of the original producing a tiny real part. Although there is clearly room for improvement here, this method has worked fine thus far.

5.2. **Data.** At this point, our tests have not been extensive enough to make any particular assertions. We have not tested enough curves to know if the early patterns we see are likely to hold in general, and we have not tested enough D's to know if we are seeing the long-term behavior for a curve, or only the initial stages of it. So, in this section, we confine ourselves to presenting some of the data, and stating our early observations.

At this stage, we have three curves with no isogenies at our disposal, plus the copious data collected by Elkies on his conductor 32 curve.

Using Mathematica, we have several tools for analyzing the data. In particular, much of the work done in this section made use Mathematica's "Fit" function. The first step, though, is just to plot the data. Throughout most of this section, the x-axis will correspond to the number of D's tested, and the y-axis will correspond to the number of rank 3 twists found. Figure 1 shows the plot for our conductor 67 curve. With the assumptions made on page 12, where we admit that we are checking only a proportion of the twists with odd rank, we would expect the plot to be on the same order as $N_{\geq 3,\mathrm{odd}}(X)$. So, we can compare our (limited) results with the conjectures.

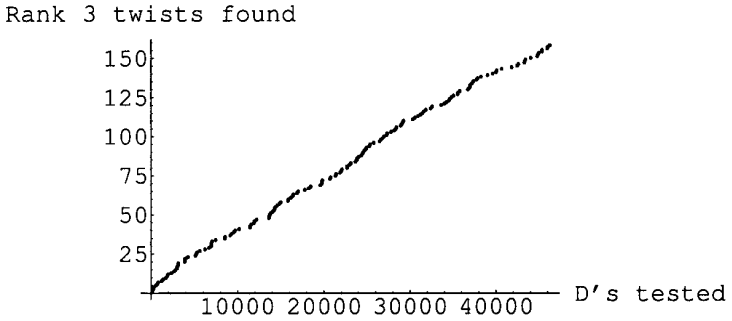

FIGURE 1. Here is a plot of the raw data for our conductor 67 curve, given by the equation $y^2 + y = x^3 + x^2 - 12x - 21$.

A cursory inspection reveals that the data looks close to linear, and fitting a linear graph (Figure 2) to the data reveals that they are indeed close. Of course, linear is a higher exponent than predicted by any of the conjectures we have considered thus far, and would contradict the density conjecture, so we are hesitant to take this first analysis too literally. Perhaps we are just not looking over enough data. For comparisons, recalling the conjectures we listed earlier, we also provide two fitted graphs which are on the order of $X^{1/4}$ (Figure 3) and $X^{3/4}$ (Figure 4). For our range, the $X^{1/4}$ estimate is not close, but the $X^{3/4}$ estimate, on the other hand, seems pretty good.

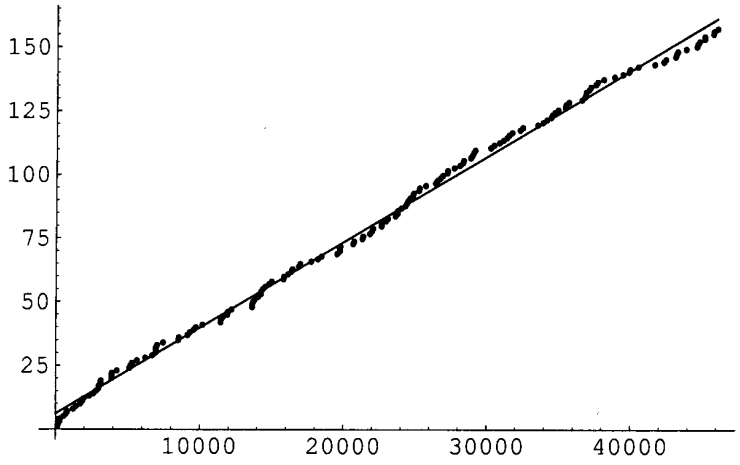

FIGURE 2. Here is our data from the conductor 67 curve, together with a line fit to the data.

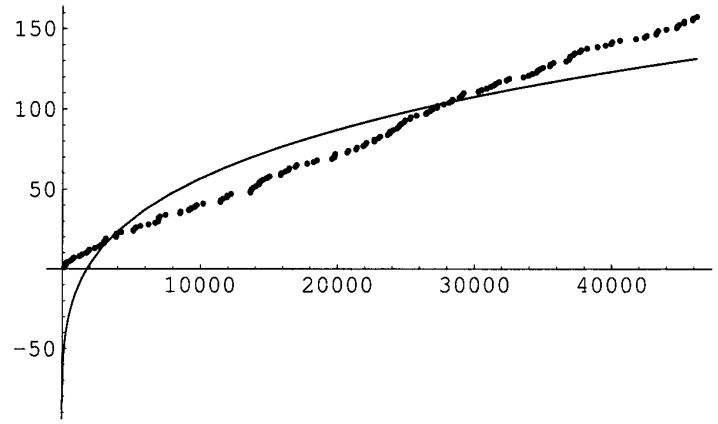

FIGURE 3. Here is our data from the conductor 67 curve, together with a fitted approximation by $X^{1/4}$.

In fact, if we let Mathematica choose the optimal power of X to fit to the data, it chooses something closer to the $X^{3/4}$ estimate than to the linear estimate. Using Mathematica's "Fit" function, together with "Exp" and "Log" commands, a graph on the approximate order of $X^{.8016}$ is chosen (Figure 5). This is comforting, as it is closer to our conjectures.

Similar calculations can be made by replacing the conductor 67 curve with the conductor 109 curve given by the equation:

$$y^2 + xy = x^3 - x^2 - 8x - 7.$$

Again, we begin with the initial plot, (Figure 6). For this data, Mathematica computes $X^{.8761}$ as a best approximation (Figure 7).

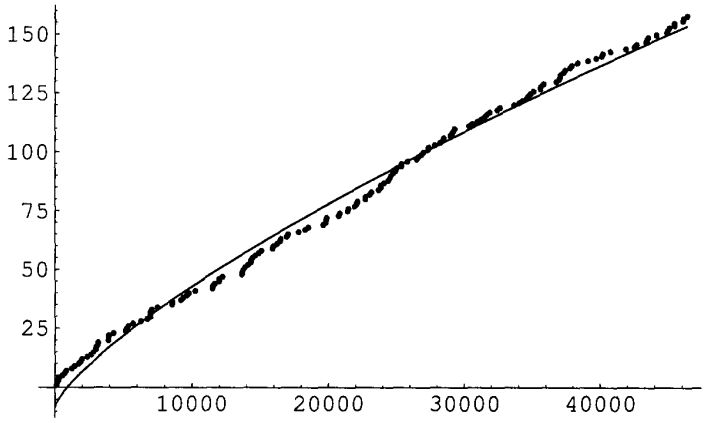

FIGURE 4. Here is our data from the conductor 67 curve, together with a fitted approximation by $X^{3/4}$.

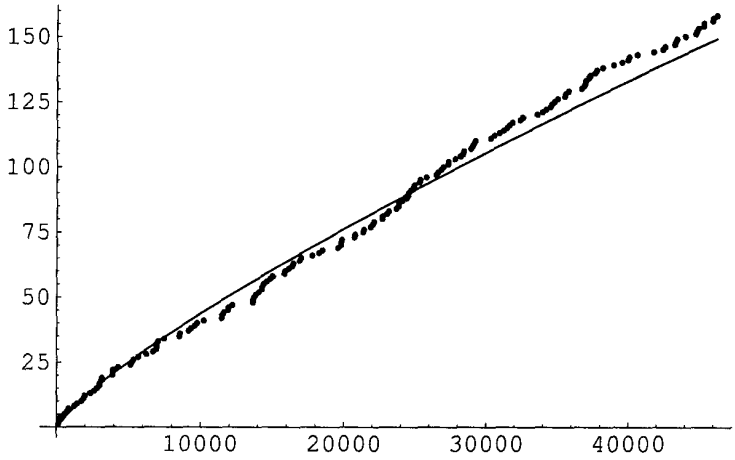

FIGURE 5. Here is our data from the conductor 67 curve, together with an $X^{.8016}$ scaled graph which Mathematica determined as a best approximation.

In both cases, the best approximation curve we come up with seems to conform to the data more closely for small D's than for large. This is obviously not what we would prefer, though, since we are interested in the asymptotic behavior of our functions. We can remedy this by merely eliminating some of the initial data we use. We include two more graphs which illustrate the effects of this, Figures 8 and 9. For our two curves, we successively ran the Mathematica "Fit" function on our collected data, but starting at a later point in the data each time. The x-axis represents our starting points, and the y-axis represents the corresponding exponent found by Mathematica. No sort of limiting process is clear from this data, but we do see the general range of exponents.

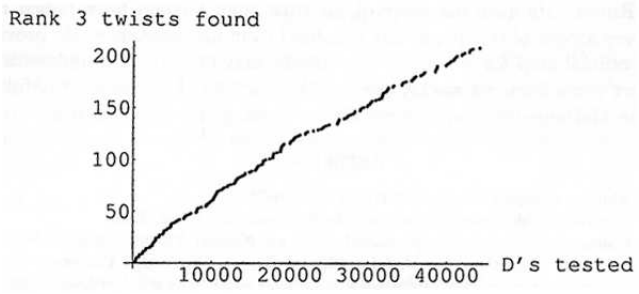

FIGURE 6. Here is the data for our conductor 109 curve.

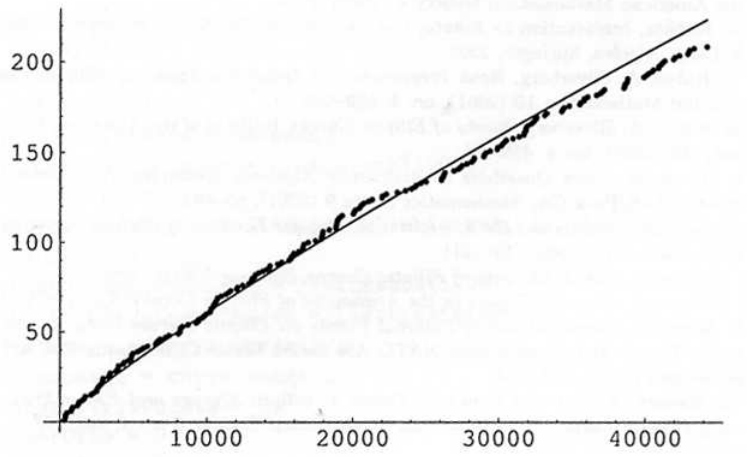

FIGURE 7. Here is our data from the conductor 109 curve, together with an $X^{.8761}$ scaled graph which Mathematica determined as a best approximation.

For comparison with a curve that has isogenies, we include a plot from Elkies data (Figure 10). Although he has much more data available than we show in our graph, we limit ourselves to similar ranges as we used for the above two curves. His data actually proves to appear significantly more linear than ours: .9781 is the initial exponent chosen by Mathematica to fit to the data, and a similar analysis to what we described above, attained by eliminating initial terms from the data, reveals the exponents all lie in the range .9 to 1.1.

An apparent anomaly (if such a term may be used in connection with our limited data) is the conductor 115 curve $y^2+y=x^3+7x-11$. The data is shown in Figure 11. Although nearly 20000 D's were tested, roughly half as many as were tested for the other two curves, only 11 rank 3 curves were identified. One notable difference between this curve and the other two with no isogenies is that its conductor has two prime factors, but our preliminary tests with the curve $106D1$ (as notated in [Cr97]) suggest that this is not the critical difference.

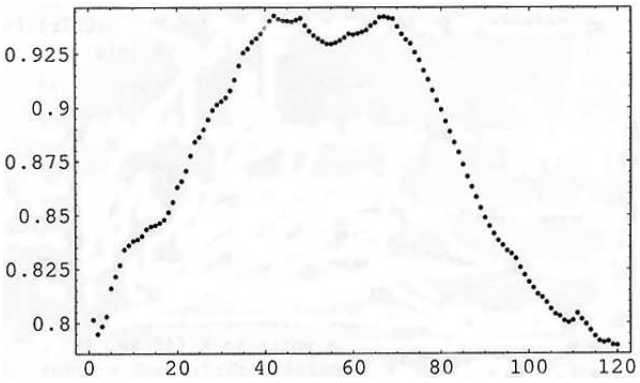

FIGURE 8. Exponent results for our conductor 67 curve.

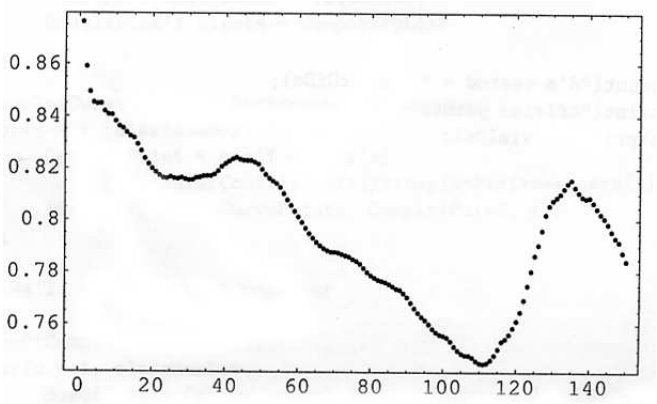

FIGURE 9. Exponent results for our conductor 109 curve.

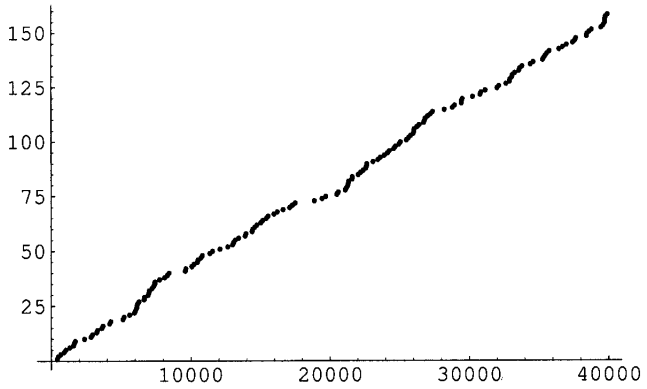

FIGURE 10. Plot from Elkies data. The x-axis here represents the number of twists *our* method would have tested, to make comparison more appropriate. (He eliminates all the remaining possibly rank 3 twists, and at least for the range we tested, our method finds the same twists as his.)

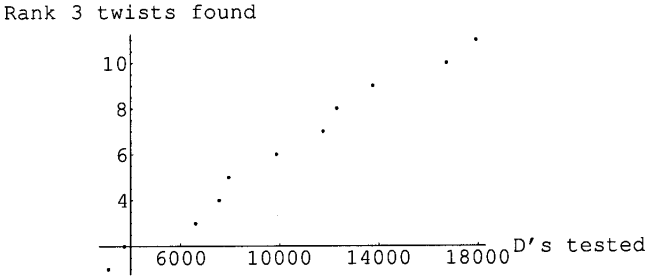

FIGURE 11. Here is the (scant) data for our conductor 115 curve.

6. Acknowledgments

This paper represents my Senior Honors Thesis, written as a senior mathematics major at Stanford University. The project was designed, supervised and supported by Karl Rubin. He gave me more of his time than I could have hoped for, and nearly every aspect of this paper has benefited from his assistance. He provided me with the crucial step for several of the proofs, and much of my understanding of the subject came from his weekly personal instruction. I am most grateful to him, and to the Mathematics Department for facilitating this opportunity.

References

[Ah79] L. Ahlfors, *Complex Analysis*, McGraw-Hill, 1979.
[BS66] Z. Borevich, I. Shafarevich, *Number Theory*, Academic Press, 1966.
[Co93] H. Cohen, *A Course in Compuational Algebraic Number Theory*, Springer-Verlag, 1993.
[Cr97] J. Cremona, *Algorithms for Modular Elliptic Curves*, Cambridge University Press, 1997. (Referenced over the internet, from www.ma.utexas.edu/ ~tornaria/cnt/cremona.html.)
[El94] N. Elkies, *Heegner Point Computations*, Algorithmic Number Theory (ANTS-1), Lecture Notes in Comp. Sci. **877** (1994), 122–133.

[GM91] F. Gouvêa, B. Mazur, *The Square-Free Sieve and the Rank of Elliptic Curves*, Journal of the American Mathematical Society **4** (1991), 1–23.
[Ko84] N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, Springer-Verlag, 1984.
[La02] S. Lang, *Algebra*, Springer, 2002.
[RS01] K. Rubin, A. Silverberg, *Rank Frequencies for Quadratic Twists of Elliptic Curves*, Experimental Mathematics **10** (2001), no. 4, 559–569.
[RS02] K. Rubin, A. Silverberg, *Ranks of Elliptic Curves*, Bulletin of the American Mathematical Society **39** (2002), no. 4, 455-474.
[Si01] A. Silverberg, *Open Questions in Arithmetic Algebraic Geometry*, Arithmetic Algebraic Geometry, IAS/Park City Mathematics Series **9** (2001), 85–142.
[Si83] J. Silverman, *Heights and the Specialization Map for Families of Abelian Varieties*, J. Reine Angew. Math. **342** (1983), 197–211.
[Si86] J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, 1986.
[Si94] J. Silverman, *Advanced Topics in the Arithmetic of Elliptic Curves*, Springer-Verlag, 1994.
[St88] N. Stephens, *Computations of Rational Points on Elliptic Curves using Heegner Points*, Number Theory and Applications, NATO ASI Series, Series C: Mathematical and Physical Sciences **265** (1988), 205–214.
[ST95] C. Stewart, J. Top, *On Ranks of Twists of Elliptic Curves and Power-Free Values of Binary Forms*, Journal of the American Mathematical Society **8** no. 4, 943–973.

Appendix: Pari Code

The following should be ready for direct input into the Pari/GP program. It represents the program as it was described in section 4, and in particular, it does not include any of the alterations we discussed elsewhere. With regards to any such alteration, though, it should be clear how to adapt the given program accordingly.

PariFile1.gp

```
{
     print("Input the elliptic curve in the usual
               [a1, a2, a3, a4, a6] format.");
     inputvector = input();
     curve = ellinit(inputvector);
     temp = ellglobalred(curve);
     n = temp[1];
     print("conductor = " n);
     m = factor(n);
     NumberOfDivisors = omega(n);
     ModularCoefficients = listcreate(10000);
     NumberOfCoefficients = 1;
     listput(ModularCoefficients, ellak(curve, 1), 1);
     ListOfPoints = [];
     \\ListOfComplexPoints = listcreate(1000);
     \\ListOfBasisCoordinates = listcreate(50000);
     firstomega = curve.omega[1];
     secondomega = curve.omega[2];
     NumberOfTrivPoints = 0;
     NumberOfDs = 0;
     TrivialDs = [];

     RememberedVariable = 5;
```

```
     maxd = 100000;

     \\The above only needs to be done once.
     \\RememberedVariable will serve as the starting
     \\point for our search for a d so that -d is a square modulo
     \\each prime divisor of n.  Each time we find a d, we'll replace
     \\RememberedVariable with d+1, so we start at the right
     \\point the next time.  maxd tells us where we'll want to stop.
     \\The program will then usually give one d above that bound.

until(RememberedVariable > maxd,

     roots = [];

     \\note below that we don't count -d as being a square
     \\mod p if -d is divisible by p.  Since we don't have the same
     \\square requirement for the prime 2, we'll input the root -1
     \\as a signal.  The 8*n term was chosen simply because n-1
     \\wasn't enough.

     for(D = RememberedVariable, maxd + 8*n,
          if(core(D) == D,
               for(x = 1, NumberOfDivisors,
                    if(m[x,1] == 2,
                         if((-D)%8 == 1%8,
                              z = 1; roots = concat(roots, -1), z = 0),
                         for(y = 1, m[x,1] - 1,
                              if((-D)%m[x,1]==y^2%m[x,1], z = 1, z = 0);
                              if(z == 1, roots = concat(roots,y);
                                   break(), z = 0);
                         );
                    );
                    if(z==0, roots = []; break(),z=1);
                    if(x==NumberOfDivisors, d = D; break(2),)
               ),
          );
     );

     RememberedVariable = d+1;
     if((-d)%4 == 1%4,
          zkSecondEl = (sqrt(-d) - 1)/2,
          zkSecondEl = sqrt(-d)
     );
     tempvar = -d;
     print("d = " tempvar);


     \\The above for loop calculates the first squarefree
```

```
\\d between RemberedVariable &
\\RememberedVariable + n-1 that is a quadratic
\\residue modulo every prime divisor of n.
\\It also creates a row vector, roots, that contains the
\\squareroots of -d modulo each prime divisor.

K = nfinit(X^2 + d);
VecIdeals = [];
for(x = 1, NumberOfDivisors,
     if((m[x,1] == 2),
          SpecialCaseIdeal = idealprincipal(K, 1);
          SpecialCaseVector = idealprimedec(K, 2);
          SpecialCaseIdeal = idealprincipal(K,
               SpecialCaseVector[1][1]);
          AnotherSpecialIdeal =
               idealprincipal(K, SpecialCaseVector[1][2][1]*K.zk[1] +
                    SpecialCaseVector[1][2][2]*K.zk[2]);
          SpecialCaseIdeal = idealadd(K, SpecialCaseIdeal,
               AnotherSpecialIdeal);
          VecIdeals = concat(VecIdeals, [SpecialCaseIdeal])
          ,
          VecIdeals = concat(VecIdeals, [idealadd(K,
               idealprincipal(K, m[x,1]),
               idealprincipal(K, Mod(roots[x] + X, X^2 + d)))])
     )
);

\\The above loop creates a vector of ideals that correspond
\\to (a + root(-d), p) for each p.  It treats the p = 2 and
\\-d=1(mod4) as a special case.  The function "idealprimedec"
\\finds the prime ideals lying above (2).

IdealIndexN = idealprincipal(K, 1);
for(x = 1, NumberOfDivisors,
     IdealIndexN = idealmul(K, IdealIndexN,
          idealpow(K, VecIdeals[x], m[x,2]))
);

\\IdealIndexN is our ideal of index N in K's ring of integers.

disc = quaddisc(-d);

classnumber = qfbclassno(disc);

if(classnumber != 1,
     classgroupinfo = quadclassunit(disc);
     classgroupstructure = classgroupinfo[2];
     classgroupgens = classgroupinfo[3];
```

```
     numbergens = length(classgroupstructure);
     ,
     classgroupstructure = [1];
     classgroupgens = [Qfb(1,1,0)];
     numbergens = 1;
);

\\The above includes the special case of trivial class group,
\\which needed to be handled differently.

IdealClassReps = listcreate(classnumber);
for(x = 1, classnumber,
     listput(IdealClassReps, 0, x)
);
partitions = 1;

if(classnumber == 1,
     listput(IdealClassReps, Qfb(1,1,0),1),
     for(x = 1, numbergens,
          partitions = partitions * classgroupstructure[x];
          partitionSize = classnumber/partitions;
          for(y = 1, partitions,
               for(z = 1, partitionSize,
                    if(IdealClassReps[(y-1)*partitionSize + z]==0,
                         listput(IdealClassReps,
                              qfbnupow(classgroupgens[x],
                                   y%classgroupstructure[x]),
                                   (y-1)*partitionSize + z),
                         listput(IdealClassReps,
                              IdealClassReps[(y-1)*partitionSize + z]
                                   *qfbnupow(classgroupgens[x],
                                   y%classgroupstructure[x]),
                                   (y-1)*partitionSize + z)
                    );
               );
          );
     );
);

\\The next step is to get a list of ideal representatives of
\\the class group.  I'm going to represent ideals as
\\matrices (representing the Z-basis) in the manner of
\\p. 220 in Cohen's book.  The basis here is different, though.
\\It's 1,root(d) if d not congruent to 1 mod 4.
\\Else it's 1, (root d - 1)/2.  I'm using the quadratic form to ideal
\\function on p. 221.  Also, note that the formula on p.221
\\expects the discriminant, so we have to multiply in the
\\case that d is 2 or 3 mod 4.
```

```
for(x = 1, classnumber,
     listput(IdealClassReps, qfbred(IdealClassReps[x]),x);
     temp = Vec(IdealClassReps[x]);
     if((-d)%4==1%4, listput(IdealClassReps, [temp[1], (1 - temp[2])/2;
                                              0, 1], x),
                     listput(IdealClassReps, [temp[1], -temp[2]/2;
                                              0, 1], x))
);

\\The list IdealClassReps will eventually hold a representative
\\from each ideal class.  To make the production easier, we
\\initialize so that each component is the trivial ideal class.
\\This way, we can recursively define things using multiplication.

\\Next we will create a list of our IJ ideals...  Here IdealIndexN
\\plays the role of I and the classreps play the roles of the J's.
\\We call the list:  SubLattices.

SubLattices = listcreate(classnumber);
for(x = 1, classnumber,
     listput(SubLattices, idealmul(K, IdealClassReps[x], IdealIndexN), x)
);

\\We now want to find the Heegner points corresponding
\\to the pairs C/IJ and J/IJ.  We have our ideals in terms of a
\\Z-basis, and our first step is actually to determine if the
\\first element in J's Z-basis is already in IJ.  Note that some
\\places want X's and others want imaginary numbers.

Heegners = listcreate(classnumber);

for(x = 1, classnumber,
     testchar = 0;
     firstcomp = IdealClassReps[x][1,1];
     secondcomp = IdealClassReps[x][2,1];
     FirstBasisElt = firstcomp + secondcomp*zkSecondEl;
     thirdcomp = IdealClassReps[x][1,2];
     fourthcomp = IdealClassReps[x][2,2];
     SecondBasisElt = thirdcomp + fourthcomp*zkSecondEl;
     TestForInclusionIdeal = idealprincipal(K,
          firstcomp + secondcomp*K.zk[2]);
     if(SubLattices[x] == idealadd(K, SubLattices[x],
              TestForInclusionIdeal),
          alpha = n * (SecondBasisElt);
          listput(Heegners, (FirstBasisElt)/alpha, x);
          testchar = 1;
     ,);
```

```
     if(testchar != 1,
          firstcompsub = SubLattices[x][1,1];
          secondcompsub = SubLattices[x][2,1];
          FirstSubBasisElt = firstcompsub +
                    secondcompsub*zkSecondEl;
          thirdcompsub = SubLattices[x][1,2];
          fourthcompsub = SubLattices[x][2,2];
          SecondSubBasisElt = thirdcompsub +
                    fourthcompsub*zkSecondEl;
          uMatrix = [firstcompsub, thirdcompsub;
                    secondcompsub, fourthcompsub];
          wMatrix = [firstcomp, thirdcomp;
                    secondcomp, fourthcomp];
          CoefMat = wMatrix^-1 * uMatrix;
          r1 = CoefMat[1,1];
          r2 = CoefMat[2,1];
          s1 = CoefMat[1,2];
          s2 = CoefMat[2,2];
          littlen = 1;
          for(y = 1, omega(n),
               if(gcd([r2, s2, m[y,1]]) == 1,
                    littlen = littlen * m[y,1];
               );
          );
          Coefs = bezout(r2 - littlen * r1, s2 - littlen * s1);
          \\print(Coefs);
          alpha = n * (FirstBasisElt + littlen * SecondBasisElt);
          listput(Heegners, (Coefs[1]*FirstSubBasisElt +
                              Coefs[2]*SecondSubBasisElt)/alpha, x);
     ,);
);

for(x = 1, classnumber,
     if(imag(Heegners[x]) < 0,
          listput(Heegners, -Heegners[x], x),
     );
);

\\The following uses an equation of p. 269 of Ahlfors to try and make
\\the imaginary part of the Heegners bigger.

print("classnumber = " classnumber);

for(x = 1, classnumber,
     RelPrimeTemp = 0;
     TempMatrix = [real(Heegners[x]*n), 1; imag(Heegners[x]*n), 0];
     TempMatrixTwo = qflll(TempMatrix);
```

```
        TempMatrix = TempMatrix * TempMatrixTwo;
        if(abs(TempMatrix[1,1] + I*TempMatrix[2,1]) <
                  abs(TempMatrix[1,2] + I*TempMatrix[2,2]),
             ctemp = n*TempMatrixTwo[1,1]; dtemp = TempMatrixTwo[2,1],
             ctemp = n*TempMatrixTwo[1,2]; dtemp = TempMatrixTwo[2,2]
        );
        if(gcd(ctemp, dtemp) == 1,
             btemp = bezout(dtemp,ctemp);
             listput(Heegners, (btemp[1]*Heegners[x] - btemp[2])/
                            (ctemp*Heegners[x] + dtemp), x),
             z = 1;
             until( 4^z > .04/abs(imag(Heegners[x])),
                  z = z + 1;
             );
             tempRepeat = z;
             NewTemp2 = 0;
             for(y = 1, tempRepeat,
                  NewTemp = 0;
                  z = 1;
                  until( (z > abs(1/imag(Heegners[x])))
                             || (NewTemp == 1) || (NewTemp2 == 1),
                       ctemp = z * n;
                       dtemp = -round(real(ctemp*Heegners[x]));
                       if( abs(ctemp*Heegners[x] + dtemp)<1
                                 && (gcd(ctemp, dtemp) == 1),
                            btemp = bezout(dtemp,ctemp);
                            listput(Heegners, (btemp[1]*Heegners[x]
                                                - btemp[2])/
                                 (ctemp*Heegners[x] + dtemp), x);
                            NewTemp = 1,
                            if(abs(ctemp * imag(Heegners[x])) > 1,
                                 NewTemp2 = 1,);
                            z = z + 1;
                       );
                  );
             );
        );
   );

   \\We now have a list "Heegners" with all the Heegner
   \\points we will need.  We will follow the Elkies article,
   \\p. 128, for mapping them initially to C modulo a lattice.
   \\We'll now find how many terms we should need to
   \\sum to put compute the Heegner points to a reasonable
   \\efficiency.  The following should be compatible with
   \\the precision gp is using.

   DesiredPrecision = 12;
```

```
ImagParts = [];

for(x = 1, classnumber,
     ImagParts = concat(ImagParts, imag(Heegners[x]));
);

MinImagPart = vecmin(ImagParts);

print(MinImagPart);

z = [];

for(x = 1, classnumber,

     z = concat(z, 50);
     while(exp(-2*Pi*z[x]*imag(Heegners[x]))/sqrt(z[x]) >
                    10^(-DesiredPrecision),
          z[x] = z[x] + 50;
     );

);

print(vecmax(z) " terms in the sum");

if(NumberOfCoefficients < vecmax(z),
     TemporaryList = listcreate(vecmax(z));
     for(x = 1, NumberOfCoefficients,
          listput(TemporaryList, ModularCoefficients[x], x);
     );
     for(x = NumberOfCoefficients + 1, vecmax(z),
          listput(TemporaryList, ellak(curve, x), x);
     );
     NumberOfCoefficients = vecmax(z);
     ModularCoefficients = TemporaryList,
);

ComplexCurvePoints = listcreate(classnumber);
for(x = 1, classnumber,
     ComplexPoint = sum(Y = 1, z[x],
            (ModularCoefficients[Y]*exp(2*Pi*I*Heegners[x])^Y)/Y, 0.);
     listput(ComplexCurvePoints, ComplexPoint, x);
);

\\We'll now sum these together.

SumOfComplexPoints = 0.;
for(x = 1, classnumber,
```

```
        SumOfComplexPoints = SumOfComplexPoints + ComplexCurvePoints[x];
    );

    print("sum = " SumOfComplexPoints);

    tempC2 = imag(SumOfComplexPoints)/imag(secondomega);
    tempC1 = (real(SumOfComplexPoints) - tempC2*real(secondomega))/firstomega;
    tempC1 = abs(tempC1 - round(tempC1));
    tempC2 = abs(tempC2 - round(tempC2));

    \\listput(ListOfBasisCoordinates, [-d, tempC1, tempC2]~, d);

    NumberOfDs = NumberOfDs + 1;

    if( (tempC1 < .0000000005) && (tempC2 < .0000000005),
        NumberOfTrivPoints = NumberOfTrivPoints + 1;
        TrivialDs = concat(TrivialDs, -d);
        write(twistfile, d " " NumberOfDs " " NumberOfTrivPoints),
    );


    \\The following can be used if we're actually interested in the
    \\coordinates.

    \\ResultingPoint = ellztopoint(curve, SumOfComplexPoints);

    \\print(ResultingPoint);

    \\The following gives an idea of whether it looks
    \\like we're really getting a point in K (if so, it should
    \\have rational real part).

    \\print(contfrac(real(ResultingPoint[1])));

);

    print("d's tested = " NumberOfDs);
    print("trivial points = " NumberOfTrivPoints);
    \\print(TrivialDs);

}
```