The elliptic curve database for conductors to 130000

John Cremona

School of Mathematical Sciences, University of Nottingham, University Park, Nottingham NG7 2RD, UK.
John.Cremona@nottingham.ac.uk

Abstract. Tabulating elliptic curves has been carried out since the earliest days of machine computation in number theory. After some historical remarks, we report on significant recent progress in enlarging the database of elliptic curves defined over $\mathbb{Q}$ to include all those of conductor $N \leq 130000$. We also give various statistics, summarize the data, describe how it may be obtained and used, and mention some recent work regarding the verification of Manin's "$c = 1$" conjecture.

1 Background and history

Tabulating elliptic curves has been carried out since the earliest days of machine computation in number theory. In this article we concentrate on tables which claim to contain complete lists of elliptic curves with conductors in certain ranges. Other tables exist, notably tables of curves with prime conductor by Brumer and McGuinness [4] and, more recently, by Stein and Watkins [21].

We first review the tables existing before 1990, and then describe the tables we have compiled since then, concentrating on the large increase in the data available since mid-2005. We will describe the origins of the tables and give some information on the methods used to compile them. We give a summary of the data obtained to date, describe how to obtain and use the data, and mention some recent work regarding the verification of Manin's "$c = 1$" conjecture.

1.1 The Antwerp Tables

For many years the only published tables giving data on elliptic curves of small conductors were those in the volume [2], popularly known as "Antwerp IV", which forms part of the Proceedings of an International Summer School in Antwerp, July/August 1972, with the title Modular Functions of One Variable IV (edited by Birch and Kuyk).

The Antwerp tables consist of the following:

Table 1: All elliptic curves of conductor $N \leq 200$, arranged into isogeny classes, with the structure of the Mordell-Weil group (in most cases) and local data for primes of bad reduction. The origin of this table is discussed below.

Table 2: Generators for the curves of positive rank (one in each isogeny class) in Table 1, which all have rank 1. These were determined independently by Nelson Stephens and James Davenport; there are two omissions (143A and 154C) and two errors (155D and 170A).

Table 3: Hecke eigenvalues for primes $p < 100$ for the newforms associated to the elliptic curves in Table 1; due to Vélu, Stephens and Tingley.

Table 4: All elliptic curves whose conductor has the form $N = 2^a 3^b$, arranged in isogeny classes (with no information on the Mordell-Weil groups); due to Coghlan.

Table 5: Dimensions of the space of newforms for $\Gamma_0(N)$ for $N \leq 300$, including the dimensions of eigenspaces for the Atkin-Lehner involutions W_q and the splitting of the space of newforms over $\mathbb{Q}$; due to Atkin and Tingley.

Table 6: Factorized polynomials in $\mathbb{F}_p[j]$, for primes $p \leq 307$, whose roots are the supersingular values of j in characteristic p; due to Atkin.

To quote [2], "The origins of Table 1 are more complicated"; two pages of [2] are devoted to explaining this further. Briefly, the list 749 curves in this table evolved as follows.

- Swinnerton-Dyer searched for curves with small coefficients and kept those with conductor $N \leq 200$; he added curves obtained via a succession of 2- and 3-isogenies. Only the coefficients, discriminant and conductor were tabulated at first.
- Higher degree isogenies were checked using Vélu's method [24], adding some curves.
- Tingley used modular symbols to compute the space of newforms for $N \leq 300$, together with the action of the Hecke algebra and hence its splitting into eigenspaces. This revealed 30 "gaps", isogeny classes which had previously been missed. These were then filled, either by twisting known curves or by extending the original search region. For example, in isogeny class $78A$ the curve with smallest coefficients is[1] $78a1 = [1, 1, 0, -19, 685]$ which is unlikely to have been found by a search. Subsequently, Tingley went on to find equations of the associated elliptic curves directly from the newforms, using a method very similar to the one which we later developed, as described in [7]. Much of Tingley's work was never published except in the contribution to the Antwerp tables, and can only be found in one of three existing typescript copies of his thesis [23] (Oxford 1975). For the higher levels in the range $N \leq 300$, Tingley's 1975 program was slow and he only computed the elliptic curves for newforms where there was no corresponding curve yet known. By contrast (and to show how both the algorithms and hardware have improved in 30 years), in 2006 our program can find these curves (for $N \leq 300$) in around 20 seconds.
- The Mordell-Weil ranks were computed by James Davenport, using the method of 2-descent as described in [3]. In eight cases these were not certain;

[1] We always specify curves by giving the coefficients $[a_1, a_2, a_3, a_4, a_6]$ of a minimal Weierstrass model. See section 3.2 below for more on labelling conventions.

in seven cases the rank is given as "0?" and is in fact 0; in one case it is given as "1?" but is 0.
- The list was known to be complete for certain conductors N, such as $N = 2^a 3^b$ and several prime N.
- Tingley's 1975 thesis [23] contains further curves with $200 < N \leq 320$ found via modular symbols, newforms and periods.

In the review by Vélu for *Mathematical Reviews* (MR0389726 (52 #10557)) a number of other minor errors in these tables are corrected.

No more systematic enumeration of elliptic curves by conductor occurred (as far as we are aware) between 1972 and the mid 1980s.

1.2 The 1992 tables

During the 1980s my research and computations mainly concerned modular symbols and elliptic curves over imaginary quadratic fields. For this, methods were developed and implemented for handling modular symbols over such fields (initially, only those of class number one), including the computation of Hecke eigenvalues and periods, and also for dealing with the easier aspects of the arithmetic of elliptic curves (conductors and point searching, but not ranks). This work included a need to have information concerning elliptic curves defined over $\mathbb{Q}$ whose conductor lay beyond the range of the Antwerp tables, which led to the development of a new implementation of the modular symbol method over $\mathbb{Q}$. At around this time, conversations with Richard Pinch led me to implement modular symbols over $\mathbb{Q}$ with quadratic character (as described in [5]).

One obstacle to the writing up of much of this work was the lack of any suitable reference in the literature to the modular symbol method over $\mathbb{Q}$ for $\Gamma_0(N)$. The new implementation was now not only recomputing from scratch all the curves listed in Antwerp IV, but also extending the list to larger conductors. Although these tables did not at this point include isogenous curves or ranks or generators, they did contain some data not in the Antwerp tables pertaining to the Birch–Swinnerton-Dyer conjectures: specifically, they contained for each curve E the rational number $L(E,1)/\Omega_E$ (where Ω_E is the least real period of E), whose value is conjectured to be 0 if and only if $E(\mathbb{Q})$ has positive rank, and is given by a conjectural formula involving the order of the Tate-Shafarevich group when $E(\mathbb{Q})$ has rank 0.

As a result, although the use of modular symbols to compute elliptic curves over $\mathbb{Q}$ was not in itself original, I decided that there was enough new material here to be worthy of publication, and in 1988 submitted a paper to *Mathematics of Computation* containing a table of elliptic curves of conductor up to 600. At this point only one curve was listed for each newform: no isogenies, ranks or generators. This paper was rejected in 1989, on two grounds: there were too many implementation details, and the referee wanted fuller information to be given for each curve – including the isogenous curves, and their ranks and generators. I was invited to resubmit the paper with this extra data included. Carrying this out required considerable effort, most significantly in re-implementing the

2-descent method of Birch and Swinnerton-Dyer to compute the ranks. James Davenport had been asked if his program for this still existed, but had replied that the only copy in existence was on a magnetic tape containing machine code for a computer which no longer existed; so this had to be done from scratch. Programs to compute isogenies and find Mordell-Weil group generators also had to be developed and written.

In 1990 I resubmitted the paper to *Mathematics of Computation*. The tables now covered all conductors to 1000, as well as containing all the requested information on ranks and generators. The text of the paper was still only 27 pages long, but it was accompanied by more than 200 pages of tables. The journal did offer to publish the paper, but with the tables as a microfiche supplement. However, while the refereeing was taking place, I was approached by several publishers who had seen the spiral-bound preprint and were interested in publishing it as a book. As nobody wanted the tables to be available only in microfiche format (which was rather old-fashioned even in 1990) I therefore withdrew it from *Mathematics of Computation* and signed up with Cambridge University Press.

Now, of course, 27 pages of text were insufficient for a book. In the first edition [6] of "Algorithms for Modular Elliptic Curves" the text was expanded to around 90 pages, with tables for curves to conductor 1000. It was published on 8 October 1992 and contained 5089 curves (those for $N = 702$ were missing through a stupid error: the number should have been 5113).

1.3 The 1997 tables

By around 1995 the book [6] was out of print and CUP asked me to prepare a revised version. This duly appeared as [7] in 1997. As well as containing corrections and the missing curves of conductor 702, some sections were rewritten and a new section and table on the degree of the modular parametrization were added. However, the range of the printed tables was not extended, though links were given to online data which extended the range to $N \leq 5077$. In addition, the period between 1992 and 1997 also saw the proof of the Shimura-Taniyama-Weil conjecture, which changed the status of some of the statements in the text as well (obviously) as the status of the tables themselves, which could now be described as listing *all* elliptic curves of conductor $N \leq 1000$ rather than just those which were modular.

The full text of [7] has been available online since around 2002.

2 Algorithms and implementation

The method we use to find all (modular) elliptic curves of a given conductor N uses modular symbols for $\Gamma_0(N)$, as is explained in detail in [7]. The original method was similar to that used by Tingley, though with certain improvements. Moreover, there have been many improvements in the details of the algorithm since the publication of [7], some of which have been developed in collaboration with William Stein. As these are rather technical we do not go into details, but

give a brief summary. For some more details (but not the more technical and recent improvements), see Chapter 2 of [7].

2.1 Finding the newforms

For each level N, one first computes the space of $\Gamma_0(N)$-modular symbols, and the action of the Hecke algebra on this space, to find one-dimensional eigenspaces with rational integer eigenvalues. Each of these corresponds to a rational newform f, where "rational" means that the Hecke eigenvalues, and hence the Fourier coefficients, are rational integers. Actually constructing the space of modular symbols is fast, though (for large levels) requires sparse matrix methods in order to fit in available machine memory. Sparse methods are also crucial when finding Hecke eigenspaces; this step is the most expensive in terms of memory requirements, and is also time-consuming, when the dimension of the space of modular symbols is large.

2.2 Finding the curves

Given the newform f, we then integrate $2\pi i f(z)dz$ along certain paths in the upper half-plane, which are also given in terms of modular symbols, to obtain first the periods and then the equation of the associated elliptic curve of conductor N and L-series $L(E,s) = L(f,s)$. Finding E in practice involves computing the period lattice of f to sufficiently high precision; which in turn requires knowing many terms of the Fourier expansion of f, i.e. many Hecke eigenvalues. From the (approximate) period lattice of E, we obtain the invariants c_4, c_6 of E, at least approximately; but they are known to be integers. [This was first made explicit by Edixhoven in [13], following Katz-Mazur; (see also [1]).] Hence c_4 and c_6 can be determined exactly if we have sufficient precision. The precision requirement means that many Hecke eigenvalues are needed (up to 3500 for levels around 130000), so for this step it is also important for the implementation to be very efficient. The memory requirements for this step, and the time to compute the periods themselves, are negligible.

2.3 Reliability of the data

Clearly no large-scale computation such as this can every guarantee 100% accuracy, and the software undoubtedly will always have bugs. Most errors to date have arisen through data processing mistakes: much of the handling of the large data files produced by our programs was done manually. More recently we have automated most of this and incorporated checks into our scripts wherever possible. Occasionally, at certain levels we missed newforms and hence elliptic curves; this has happened most often just after major rewriting of the code. When curves are missed at level N, we usually discover the fact when processing level $2N$, since then certain oldforms are not recognised as such. The online data is updated regularly and such corrections are logged; the data imported into packages (see below) may not be quite so up-to-date.

2.4 Obtaining information about the curves

For each elliptic curve found, we determine the analytic rank from the newform; when this is greater than 1 we check that it equals the Mordell-Weil rank using 2-descent.

Generators are found using a combination of the traditional methods: (1) search; (2) 2-descent, using our program `mwrank` [10]; (3) Heegner points (we now use MAGMA [16] for these, as the current implementation by Watkins, based on earlier versions by Cremona, Womack, Watkins and Delaunay, is extremely efficient); plus saturation methods.

We also compute isogenies, and all data on the isogenous curves. Since the computation of isogenies is rather delicate (it is easy to miss some if the precision is insufficient) this is done independently, as a check, using a program of Mark Watkins; as a benefit, Watkins's program also computes the degree of the modular parametrization and determines the curve in each isogeny class of minimal Faltings height. This method of computing the modular degree (described in [25]) is very much more efficient than the original one described in [8], which we stopped using at around $N = 14000$. The Faltings height information also allows verification of Stevens's Conjecture [22], that the curve with minimal Faltings height in each isogeny class is the one associated with $\Gamma_1(N)$ (which is usually, though not always (especially for smaller N), the same as the curve associated with $\Gamma_0(N)$).

2.5 Software

The original program was written in the 1980s in `Algol68`, and converted to `C++` in the early 1990s. We use either Victor Shoup's `NTL` library (see [15]) or the `LiDIA` library (see [14]) for high-precision arithmetic, as well as `STL` (the Standard Template Library for `C++`). The sparse matrix code has been completely redeveloped, based on an earlier version by Luiz Figueiredo. This is probably the most important single programming improvement, and is essential both to physically allow levels as high as 100000 to be run on a machine with 2GB of RAM, and also for greatly increased speed of execution. Even so, some levels around 130000 require more than 2GB of RAM in which to run.

Without many low-level efficiency and algorithmic improvements it would not have been possible to have progressed so far. Some of these have been developed in collaboration with William Stein, who has written more general programs for computing with higher weights and characters: implemented originally in `C++`, then in MAGMA, and most recently in his package `SAGE` (see [18] and [19]).

One example: in [21] an example is given of a curve of rank 2 and rational 5-torsion of conductor 13881, which was then (2002) "beyond the range of Cremona's tables"; computing the four curves (up to isogeny) of this conductor now takes less than 2 minutes to run, requiring about 60MB of RAM. Most of the computation time is taken up finding the eigenspaces for the first Hecke operator T_2 on the modular symbol space of dimension 1768.

2.6 Hardware

The other factor which has had an enormous impact on the expansion of the tables since spring 2005 is the availability at the University in Nottingham of a 1024-processor High Performance Computing "GRID" cluster, on which each user may (normally) use up to 256 processors simultaneously. This has enabled the processing of a hundred or more levels at a time. The GRID processors are arranged in pairs in 512 nodes, with each node (a "V20z dual Opteron") having access to its own 2GB of RAM. No parallel code is used (yet), so the advantage of the cluster is simply that of having a large number of machines controlled via a scheduling system to keep them all busy with the minimum amount of human intervention.

The nodes in the cluster have "only" 2GB of RAM each; hence for some larger levels it is necessary to perform separate runs on a different machine, with more RAM (8GB). So far this has sufficed, but further developments in the code are under way to enable the current upper bound of 130000 to be passed.

2.7 Milestones

Before using the HPC GRID we used between 0 and 3 machines, all shared with other users and jobs.

Date	Conductor reached
Mar 2001	10000
Nov 2001	12000
Aug 2002	13000
Oct 2002	15000
Jan 2003	16000
Feb 2003	18000
Mar 2003	19000
Apr 2003	20000
Mar 2004	21000
Apr 2004	23000
May 2004	24000
Jun 2004	25000
Oct 2004	26000
Nov 2004	27000
Jan 2005	29000
Feb 2005	30000

After starting to use the HPC GRID, the pace increased considerably:

Date	Conductor reached
22 Apr 2005	40000
27 May 2005	50000
9 Jun 2005	60000
20 Jun 2005	70000
14 Jul 2005	80000
26 Aug 2005	90000
31 Aug 2005	100000
18 Sep 2005	120000
3 Nov 2005	130000

Currently the program is undergoing further refinements in the expectation that it will be able to make further progress without moving wholesale to machines with more RAM. It would be interesting to cover all levels to $N = 234446$, which is the smallest known conductor of a curve with rank 4, namely $234446b1 = [1, -1, 0, -79, 289]$. Level $N = 234446$ itself has been run successfully; as well as the rank 4 curve there are two others with this conductor, both of which have rank 3: $234446a1 = [1, 1, 0, -696, 6784]$ and $234446c1 = [1, 1, 1, -949, -7845]$.

2.8 Using the GRID

To use the HPC GRID we use a fairly simple shell script, which loops over a range of values of N. This script runs simultaneously on however many nodes are available. At each pass through the loop, shell commands are used to detect the existence of a log file associated with the value of N in question, which would indicate that another node was already working on this level. If so, N is incremented; otherwise a series of `C++` programs is run with N (and other parameters) as input, which result in all the necessary computations being carried out for that level with the output suitably recorded. One minor technical issue here is that the system has to be able to handle several hundreds of thousands of data files, something of which system administrators may disapprove. [We keep the data for each level accessible for later runs, since our method of eliminating oldforms currently involves accessing the data at levels M dividing N, rather than using degeneracy maps.]

A typical extract from the log file of one node follows:

```
running nfhpcurve on level 120026 at Fri Sep 23 18:26:48 BST 2005
running nfhpcurve on level 120197 at Fri Sep 23 20:12:31 BST 2005
running nfhpcurve on level 120224 at Fri Sep 23 20:58:18 BST 2005
running nfhpcurve on level 120312 at Fri Sep 23 23:35:19 BST 2005
running nfhpcurve on level 120431 at Sat Sep 24 04:19:54 BST 2005
running nfhpcurve on level 120568 at Sat Sep 24 10:42:18 BST 2005
running nfhpcurve on level 120631 at Sat Sep 24 13:56:49 BST 2005
running nfhpcurve on level 120646 at Sat Sep 24 14:48:21 BST 2005
running nfhpcurve on level 120679 at Sat Sep 24 15:59:54 BST 2005
running nfhpcurve on level 120717 at Sat Sep 24 18:11:20 BST 2005
```

```
running nfhpcurve on level 120738 at Sat Sep 24 19:13:11 BST 2005
running nfhpcurve on level 120875 at Sun Sep 25 02:20:27 BST 2005
running nfhpcurve on level 120876 at Sun Sep 25 02:20:28 BST 2005
running nfhpcurve on level 120918 at Sun Sep 25 04:58:32 BST 2005
running nfhpcurve on level 120978 at Sun Sep 25 08:08:00 BST 2005
```

The program being run here is called "`nfhpcurve`", where "`nf`" stands for newform, "`hp`" for "H_1^+" indicates that we use the plus part of the modular symbol space, and "`curve`" that we compute the equations for the curve from each newform. Separate programs are run to find isogenous curves and Mordell-Weil generators and other data.

The levels here are in the range 120000–121000; those not listed are being run on other nodes. Approximately 10 levels per processor per day are completed, though the time for each individual level varies greatly, depending on several factors: highly composite N have modular symbol spaces of higher dimension, which has a major effect on the time required for linear algebra; levels with no newforms obviously save on the time required to compute many Hecke eigenvalues a_p; and curves with very large c_4, c_6 invariants require working to higher precision with more a_p needing to be computed.

Certain values of N are known not to be possible conductors (specifically, N which are divisible by 2^9 or by 3^6 or by p^3 with $p \geq 5$) and these are skipped.

3 Summary of data and highlights of results

3.1 Availability of the data

Full data is available from [9]. The data is mostly in plain `ascii` files for ease of use by other programs, rather than in typeset tables as in the book. A mirror is maintained by William Stein at `http://modular.math.washington.edu/cremona/`. (Stein's Modular Forms Database at `http://modular.math.washington.edu/Tables/` also has links to many other tables.) Currently there is approximately 106MB of data (as a gzipped tar file) which unpacks to 260MB. This only includes a_p for $p < 100$, as further values can obviously be recomputed from the curve itself.

Recently, in collaboration with various other people, other more convenient ways of accessing and processing the data have been developed.

- A web-based interface by Gonzalo Tornaria is at `http://www.math.utexas.edu/users/tornaria/cnt/cremona.html`, covering $N < 100000$. This provides an attractive interactive interface to the data; as a bonus, information on quadratic twists is included.
- The free open-source number theory package `pari/gp` (see [17]) makes the full elliptic curve database available (though not installed by default). For example

```
(12:05) gp > ellsearch(5077)
```

```
%1 = [["5077a1", [0, 0, 1, -7, 6], [[-2, 3], [-1, 3], [0, 2]]]]
(12:05) gp > ellinit("5077a1")
%2 = [0, 0, 1, -7, 6, 0, -14, 25, -49, 336, -5400, 5077, ...
(12:05) gp > ellidentify(ellinit([1,2,3,4,5]))
%3 = [["10351a1", [1, -1, 0, 4, 3], [[2, 3]]], [1, -1, 0, -1]]
```

The output of `ellsearch` contains all matching curves with their generators. The output of `ellidentify`, whose input need not be given in minimal or standardised form, includes the standard transformation $[u, r, s, t]$ mapping the input curve to standard minimal form. Full integration of this capability with standard `pari/gp` elliptic curve functions is ongoing (thanks to Bill Allombert).

– William Stein's free open-source package `SAGE` (Software for Algebra and Geometry Experimentation, see [18] and [19]) also has all our data available and many ways of working with it, including a transparent interface to many other pieces of elliptic curve software. For example:

```
sage: E = EllipticCurve("389a")
sage: E
Elliptic Curve defined by y^2 + y = x^3 + x^2 - 2*x  over Rational Field
sage: E.rank()
2
sage: E.gens()                              # Cremona's mwrank
[(-1 : 1 : 1), (0 : 0 : 1)]
sage: L = E.Lseries_dokchitser(); L(1+I)   # Tim Dokchitser's program
-0.63840993858803874 + 0.71549523920466740*I
sage: E.Lseries_zeros(4)                    # Mike Rubinstein's program
[0.00000000000, 0.00000000000, 2.8760990715, 4.4168960843]
```

– Magma has the database for conductors up to 70000 (as of version 2.12-16):

```
> ECDB:=CremonaDatabase();
> NumberOfCurves(ECDB);
462968
> LargestConductor(ECDB);
70000
> E:=EllipticCurve(ECDB,"389A1");
> E;
Elliptic Curve defined by y^2 + y = x^3 + x^2 - 2*x over Rational Field
> Rank(E);
2
```

3.2 The naming of curves

Since many authors refer to individual elliptic curves by means of their label in the database, it is desirable to use a sensible naming convention which is concise, informative and only changes when absolutely necessary.

The Antwerp tables use a labelling system for the elliptic curves which consists of the conductor followed by a single upper case letter. The order of these is not easy to define; the curves are grouped into isogeny classes, but one cannot determine this from the label alone. For example, the curves of conductor 37 are in two classes, $\{37A\}$ and $\{37B, 37C, 37D\}$. Clearly this system cannot be used once we have more than 26 curves per conductor.

For the tables of [7] we introduced an additional layer into the notation. The isogeny classes have labels similar to those of individual curves in the Antwerp system, consisting of a single uppercase letter following the conductor. The curves in the class are indicated by suffixing (or occasionally subscripting) the class code with an integer. For example at conductor 37 the classes are $\{37A1\}$ and $\{37B1, 37B2, 37B3\}$.

The ordering of the isogeny classes is determined by the order in which the newforms are found with our modular symbols program; this has changed over the years and so is now, unfortunately, almost impossible to define precisely. However for all levels between 451 and 130000 the order is lexicographical order of the Hecke eigenvalues of the newforms, with the eigenvalues of the Atkin-Lehner involutions W_q (for bad primes q) listed first, and the eigenvalues for W_q ordered $+1, -1$ those for T_p as $0, +1, -1, +2, -2, \ldots$. It is planned to change this system for $N > 130000$ to one based on simple lexicographical order of the complete eigenvalue sequence, with all primes in their natural order; but there will be no further change in the labels for $N \leq 130000$!

The order of the curves within each isogeny class is likewise difficult to define precisely. The first curve in each class is the curve variously called the "strong Weil curve" or the $\Gamma_0(N)$-optimal curve; that is, the curve whose period lattice (of a minimal model) is exactly that of the normalised newform. After that, the order is determined by our algorithm for finding isogenies.

In the tables in [7], for $N \leq 200$ the Antwerp codes were given alongside the new ones.

When the tabulation reached $N = 1728$, where there are for the first time more than 26 isogeny classes (there are 28), something new was required. Without sufficient thought for "future-proofing" we simply followed the sequence $A, B, \ldots, Z$ by $AA, BB, \ldots, ZZ$ and then (at level $N = 4800$ which has 72 rational newforms) $AAA, BBB, \ldots$ and so on. In 2005 this system was becoming unworkable. At level 100800 there are 418 rational newforms with codes from $100800A$ to $100800BBBBBBBBBBBBBBBBBBBBBBBBBB$.

It was therefore decided to use a new coding system for the isogeny class labels, and after widespread consultation the following scheme was decided upon (thanks to David Kohel in particular). We now use a base 26 number system, with the letters $a, \ldots, z$ for the "digits" $0, \ldots, 25$ and leading as omitted. So after z comes ba, and the last class at level 100800 has label $100800qb$. For conductor 37, the classes are now $\{37a1\}$ and $\{37b1, 37b2, 37b3\}$. When we reach a conductor where the number of classes is more than $26^2 = 676$, all we need do is follow zz with baa. [In the Stein-Watkins database of elliptic curves there are conductors with many thousands of isogeny classes.]

Lower case letters were used to avoid confusion between old and new coding systems; so (happily) the only difference for curves of conductor less than 1728 is the change of case.

The online tables have been altered to reflect this change of coding, as have the databases available in `SAGE` and `pari/gp`, but MAGMA V2.12 still uses the old system.

3.3 Numbers of curves

In Table 1 we give the numbers of isogeny classes of curves for ranges of conductors of the form $10000k \leq N < 10000(k+1)$, together with the numbers for each value of the rank. One very remarkable feature is that the number in each range is close to constant. This feature is maintained in smaller ranges: in each range of 1000 consecutive conductors there are very close to 4400 isogeny classes of curves.

range of N	#	$r=0$	$r=1$	$r=2$	$r=3$
0-9999	38042	16450	19622	1969	1
10000-19999	43175	17101	22576	3490	8
20000-29999	44141	17329	22601	4183	28
30000-39999	44324	16980	22789	4517	38
40000-49999	44519	16912	22826	4727	54
50000-59999	44301	16728	22400	5126	47
60000-69999	44361	16568	22558	5147	88
70000-79999	44449	16717	22247	5400	85
80000-89999	44861	17052	22341	5369	99
90000-99999	43651	16370	21756	5442	83
100000-109999	44274	16599	22165	5369	141
110000-119999	44071	16307	22173	5453	138
120000-129999	44655	16288	22621	5648	98
0-129999	568824	217401	288675	61840	908

Table 1. Numbers of isogeny classes of curves, by rank

The chart in Figure 1 shows the overall distribution of ranks.

In Table 2 we give the total number of curves up to isomorphism. This reveals that the average size of the isogeny classes found is currently about 1.487. This average seems to be steadily but gradually decreasing (the value for $N \leq 1000$ was just over 2.0). Mark Watkins has pointed out that if one considers curves in a large box with $|c_4| < X^2$ and $|c_6| < X^3$, then the average size of the isogeny class tends to 1 as $X \to \infty$. Also, Duke has shown in [12] that almost all curves

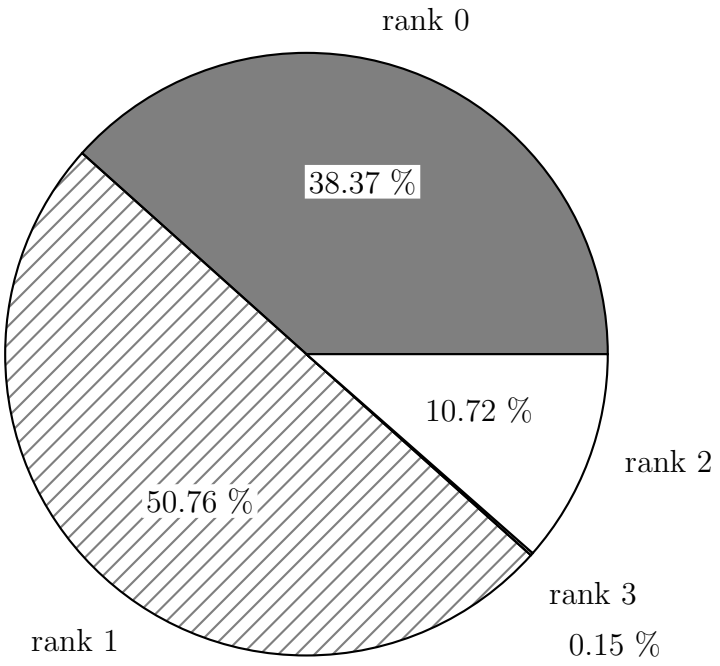

Fig. 1. Overall distribution of ranks

(ordered in this way) have no exceptional primes, and in particular no rational isogenies.

The sizes of individual isogeny classes are given in Table 3. Here we classify isogeny classes by the maximal degree D of an isogeny (with cyclic kernel) between curves in the class. For each possible value of D, there is a uniquely determined shape of the graph of curves and isogenies of prime degree between them. (See Table 1 of Antwerp IV for examples of most of these.)

3.4 Mordell-Weil groups

For almost all the elliptic curves found we have determined the full Mordell-Weil group. In a very small number of cases we can only (at present) guarantee that the generators listed in the tables generate a subgroup of finite index. In all cases where the analytic rank is 2 or 3 we have verified by 2-descent that the rank is equal to the analytic rank. When the analytic rank is 0 or 1 this is known to be true by results of Rubin, Kolyvagin and Gross and Zagier.

In most cases of positive rank, searching for points suffices to find the expected number of independent generators, following which we apply a saturation procedure to obtain the full Mordell-Weil group. The exceptional cases are those for which we were not able to determine a bound on the index, on account of the bound on the difference between the logarithmic and canonical heights be-

range of N	# isogeny classes	# isomorphism classes
0-9999	38042	64687
10000-19999	43175	67848
20000-29999	44141	66995
30000-39999	44324	66561
40000-49999	44519	66275
50000-59999	44301	65393
60000-69999	44361	65209
70000-79999	44449	64687
80000-89999	44861	64864
90000-99999	43651	63287
100000-109999	44274	63410
110000-119999	44071	63277
120000-129999	44655	63467
0-129999	568824	845960

Table 2. Numbers of isogeny and isomorphism classes of curves

D	Size	# classes	%
1	1	372191	65.43
2	2	123275	21.67
3	2	31372	5.52
4	4	27767	4.88
5	2	2925	0.51
6	4	3875	0.68
7	2	808	0.14
8	6	2388	0.42
9	3	2709	0.48
10	4	271	0.05
11	2	60	0.01
12	8	286	0.05
13	2	130	0.02

D	Size	# classes	%
14	4	28	< 0.01
15	4	58	0.01
16	8	270	0.05
17	2	8	< 0.01
18	6	162	0.03
19	2	12	< 0.01
21	4	30	0.01
25	3	134	0.02
27	4	33	0.01
37	2	20	< 0.01
43	2	7	< 0.01
67	2	4	< 0.01
163	2	1	< 0.01

Table 3. Distribution of isogeny class sizes and degrees

ing rather large. This situation should improve after full implementation of the improved height bound algorithm described elsewhere in this volume: see [11].

For curves where searching for points was insufficient, most had rank 1 and generators could be found using Heegner points. Since 2004, techniques for computing Heegner points of large height have improved very significantly, thanks to work of Delaunay and Watkins. The MAGMA implementation is now extremely fast and we have used it extensively for all the larger generators in the tables.

The current record is curve $108174c2$, whose generator P has canonical height $\hat{h}(P) = 1193.35$. Here $P = (a/c^2, b/c^3)$ where

$$a = -136328337031406810335030236791286705295582184200634323979714392818761689369256080992786861037682711657514376335562130410241362759901574725088011823024544366789004558603070348135761058684475116028333276569784624255741311649448653831044747619035843993306071711117602972355733099941007766410489359701348123605207598742554713521099294186837422237009896297109549762937178684101535289410605736729335307780613198224770325365111296070756137349249522158278253743039282375024853516001988744749085116423499171358836518920399114139315005,$$

$$b = 776845386159678589635077615346492181601035042768002014396646962333772688446303892162606526955979081249211185106671917236143678971202347339963247386055808925185619325909681380265508543158979491984235466881248491978341526711100575326744746030922470291782156359389005809065313914236892470866399096616908015986267206085816145609347461468770147859622405813347969542380216159923828490925517451952455079424426512616714569247069065790676549942365146817589522964032348349807255751358289869629122053879780510640219504941970766697032823589255263953926885142009701275092664710953135501372398976396568319085695054751879368605289437600720585853465424006259176930980665902501637183477157293942231705607887213321716750749368884791336280387610317598902033025432647703668271483782740137711508479669 1,$$

$$c = 113966855669333292896328833690552943933212422262287285858336471843279644076647486592460242089049033370292485250756121056680073078113806049657487759641390843477809887412203584409641844116068236428572188929747769498615000931961765366269300665024812605970444134 7.$$

In MAGMA, finding this generator is as easy (and quick) as this:

```
> E:=EllipticCurve([1,1,0,-330505909530535,-2312687660697986706251]);
> time HeegnerPoint(E);
true (-13632833.../12988444... : 77684538.../14802521... : 1)
Time: 36.340
```

At the other extreme, the minimal height of a generator is for curve $3990v1 = [1, 1, 1, -125615, 61201397]$, whose generator $(7107, -602054)$ has canonical height 0.0089.

In the small number of cases where curves of rank greater than 1 have generators too large to be found easily by searching, we found the generator using 2-descent and in some cases 4-descent. The latter, for which algorithms were developed by Siksek and Womack, is now efficiently implemented in MAGMA.

3.5 Torsion structures

The distribution of the 15 possible structures for the torsion subgroups of the curves is given in Table 4. Here C_n denotes a cyclic group of order n.

Structure	# curves	%
C_1	432622	51.14
C_2	344010	40.67
C_3	18512	2.19
C_4	12832	1.52
$C_2 \times C_2$	33070	3.91
C_5	698	0.08
C_6	3155	0.37
C_7	50	< 0.01
C_8	101	0.01
$C_2 \times C_4$	793	0.09
C_9	16	< 0.01
C_{10}	28	< 0.01
C_{12}	11	< 0.01
$C_2 \times C_6$	58	< 0.01
$C_2 \times C_8$	4	< 0.01
Odd	451898	53.42
Even	394062	46.58
All	845960	100.00

Table 4. Torsion structures

3.6 Degrees of modular parametrizations

As already mentioned, for most of the range the modular parametrization degrees were computed using a program of Mark Watkins (see [25] for the method). Here we only mention that the largest degree so far is for $96054k1$, for which $\deg(\varphi) = 32035843840 = 2^8 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 2273$.

3.7 (Analytic) orders of Ш

For each curve E in the tables, we have computed all the quantities appearing in the Birch–Swinnerton-Dyer conjecture for E, with the exception of the order of the Tate-Shafarevich group Ш. It is customary to define the "analytic order of Ш" to be the order predicted by the Birch–Swinnerton-Dyer conjecture, which we can determine from this data. In the case of curves of rank 0 this is computed as an exact rational number, which turns out in every case to be an integral perfect square. For curves of positive rank it is computed as a floating-point approximation (using approximations for the regulator, the real period and $L^{(r)}(E,1)$); we again always find a value close to an integer which is a perfect square. In Table 5 below, we do not distinguish between the different status of these values. The current record is $676 = 26^2$ for curve $95438a1$ (which has rank 0 so this is an exact value).

$\lvert Ш\rvert$	#
2^2	37074
3^2	11512
4^2	4013
5^2	1954
6^2	426
7^2	468
8^2	250
9^2	85
10^2	52
11^2	73
12^2	20
13^2	19

$\lvert Ш\rvert$	#
14^2	9
15^2	2
16^2	6
17^2	4
19^2	2
20^2	3
21^2	2
23^2	4
26^2	1
all > 1	55979

Table 5. Analytic orders of Ш

We should also mention here recent work of Stein and others (see [20]) towards verifying precisely the Birch–Swinnerton-Dyer conjecture for non-CM curves of rank at most 1 and conductor up to 1000.

3.8 The Manin constant

Recall that the Manin constant for an elliptic curve E of conductor N is the rational number c such that

$$\varphi^*(\omega_E) = c(2\pi i f(z)dz),$$

where ω_E is a Néron differential on E, f is the normalized newform for $\Gamma_0(N)$ associated to E, and $\varphi : X_0(N) \to E$ is the modular parametrization. A long-standing conjecture is that $c = 1$ for all elliptic curves over $\mathbb{Q}$ which are optimal quotients of $J_0(N)$ (or "strong Weil curves" in the older terminology). A result already cited [13] is that $c \in \mathbb{Z}$, and there are many results restricting the primes which may divide c.

Recent developments, described in [1], have strengthened these conditions considerably. Also in [1] there is an account of numerical verifications we have carried out which establish the conjecture for most of the curves in our tables. The following result is taken from [1].

Theorem 1. *(a) For all $N \leq 60000$, every optimal elliptic quotient of $J_0(N)$ has Manin constant equal to 1.*

(b) For all N in the range $60000 < N \leq 130000$, every optimal elliptic quotient of $J_0(N)$ has Manin constant equal to 1, except[2] for the following cases where the Manin constant is either 1 or 2:

$$67664a, 71888e, 72916a, 75092a, 85328d, 86452a, 96116a,$$
$$106292b, 111572a, 115664a, 121168e, 125332a.$$

In each of the 12 undecided cases listed, the isogeny class consists of two curves linked by 2-isogenies, and we have not yet verified which of the two curves is the optimal quotient of $J_0(N)$. See [1] for details.

Acknowledgements

Thanks to William Stein and Mark Watkins for comments on an earlier draft of this paper.

References

1. A. Agashe, K. Ribet and W. A. Stein (with an appendix by J. E. Cremona), *The Manin Constant, Congruence Primes and the Modular Degree*, preprint 2006.
2. B. J. Birch and W. Kuyk (eds.), *Modular Functions of One Variable IV*, Lecture Notes in Mathematics **476**, Springer-Verlag 1975. See also `http://modular.math.washington.edu/scans/antwerp/`.
3. B. J. Birch and H. P. F. Swinnerton-Dyer, *Notes on Elliptic Curves I*, J. Reine Angew. Math. **212** (1963), 7–25.
4. A. Brumer and O. McGuinness, *The behaviour of the Mordell-Weil group of elliptic curves*, Bull. Amer. Math. Soc. (N.S.) **23** (1990), no. 2, 375–382.

[2] Note added in proof: In all the cases listed we have now verified that $c = 1$.

5. J. E. Cremona, *Modular symbols for $\Gamma_1(N)$ and elliptic curves with everywhere good reduction*, Math. Proc. Cambridge Philos. Soc., **111** (1992) no. 2, 199–218.
6. J. E. Cremona, *Algorithms for modular elliptic curves*, Cambridge University Press, 1992.
7. J. E. Cremona, *Algorithms for modular elliptic curves (2nd edition)*, Cambridge University Press, 1997. See also `http://www.maths.nott.ac.uk/personal/jec/book/fulltext/`.
8. J. E. Cremona, *Computing the degree of the modular parametrization of a modular elliptic curve*, Math. Comp. **64** no. 211 (1995), 1235–1250.
9. J. E. Cremona, Tables of Elliptic Curves, `http://www.maths.nott.ac.uk/personal/jec/ftp/data/`.
10. J. E. Cremona, `mwrank`, a program for 2-descent on elliptic curves over $\mathbb{Q}$, `http://www.maths.nott.ac.uk/personal/jec/mwrank/`.
11. J. E. Cremona and S. Siksek, *Computing a Lower Bound for the Canonical Height on Elliptic Curves over* $\mathbb{Q}$, in F.Hess, S.Pauli and M.Pohst (eds.), ANTS 2006, Lecture Notes in Computer Science 4076 (2006).
12. W. Duke, *Elliptic curves with no exceptional primes*, C. R. Acad. Sci. Paris Sér. I Math. **325** (1997), no. 8, 813–818.
13. B. Edixhoven , *On the Manin constants of modular elliptic curves*, in *Arithmetic algebraic geometry (Texel, 1989)*, Progr. Math. **89**, Birkhäuser, Boston (1991), 25–39.
14. `LiDIA`: A `C++` Library For Computational Number Theory, `http://www.informatik.tu-darmstadt.de/TI/LiDIA/`.
15. `NTL`: A Library for doing Number Theory, `http://www.shoup.net/ntl/`.
16. The MAGMA Computational Algebra System, version 2.12-16, `http://magma.maths.usyd.edu.au/magma/`.
17. `pari/gp`, version `2.2.13`, Bordeaux (2006), `http://pari.math.u-bordeaux.fr/`.
18. W. A. Stein and D. Joyner, `SAGE`*: System for Algebra and Geometry Experimentation*, Comm. Computer Algebra **39** (2005), 61–64.
19. W. A. Stein, `SAGE`, Software for Algebra and Geometry Experimentation, `http://sage.scipy.org/sage`.
20. W. A. Stein, G. Grigorov, A. Jorza, S. Patrikis and C. Patrascu, *Verification of the Birch and Swinnerton-Dyer Conjecture for Specific Elliptic Curves*, preprint 2006.
21. W. A. Stein and M. Watkins, *A database of elliptic curves—first report*, in *Algorithmic number theory (Sydney, 2002)*, Lecture Notes in Comput. Sci., **2369**, Springer, Berlin, 2002, 267–275.
22. G. Stevens, *Stickelberger elements and modular parametrizations of elliptic curves*, Invent. Math. **98** (1989), no. 1, 75–106.
23. D. J. Tingley, *Elliptic curves uniformized by modular functions*, University of Oxford D. Phil. thesis (1975).
24. J. Vélu, *Isogénies entre courbes elliptiques*, C. R. Acad. Sci. Paris Sér. A-B, **273** (1971), A238–A241.
25. M. Watkins, *Computing the modular degree of an elliptic curve*, Experiment. Math. **11** no. 4 (2002), 487–502.