manuscripta math. 95, 463 – 469 (1998)

manuscripta mathematica
© Springer-Verlag 1998

A procedure to calculate torsion of elliptic curves over $\mathbb{Q}$

Darrin Doud

Department of Mathematics, University of Illinois, Urbana, IL 61801, USA.
e-mail: doud@math.uiuc.edu

Received: 7 August 1997 / Revised version: 28 November 1997

Abstract. We present an algorithm which uses the analytic parameterization of elliptic curves to rapidly calculate torsion subgroups, and calculate its running time. This algorithm is much faster than the "traditional" Lutz–Nagell algorithm used by most computer algebra systems to calculate torsion subgroups.

Mathematics Subject Classification (1991): 11G05

1. Elliptic curves

An elliptic curve over $\mathbb{Q}$ is the set of solutions to an equation of the form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where the a_i are rational numbers. For K a field containing $\mathbb{Q}$, we will denote the solutions which have coordinates in K by $E(K)$. It is well known that for K a number field, $E(K)$ is a finitely generated abelian group, hence it has a finite torsion subgroup. In calculating the torsion subgroup, of $E(\mathbb{Q})$, we will generally use a change of coordinates to put the equation defining the elliptic curve into one of two simpler forms, either $y^2 = 4x^3 + Ax + B$ or $y'^2 = x'^3 + 4Ax' + 16B$, with A and B integers. We note that these two forms are related via $x' = 4x$, $y' = 4y$. The discriminant of the elliptic curve is given for a curve in the first form by the formula $-A^3 - 27B^2$; for the second form, the discriminant is 2^{12} times this value.

The following theorem, due to Mazur, limits the groups which can occur as rational torsion subgroups:

Theorem 1 (Mazur). *The torsion group of any elliptic curve over $\mathbb{Q}$ is isomorphic to $\mathbb{Z}/n\mathbb{Z}$ for $1 \leq n \leq 10$ or $n = 12$, or $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2n\mathbb{Z}$ for $1 \leq n \leq 4$.* ([5], p. 223.)

2. The theorem of Lutz–Nagell

The following theorem is the basis for the most commonly used algorithm for computing rational torsion subgroups:

Theorem 2 (Lutz–Nagell). *Given any elliptic curve with coefficients in $\mathbb{Z}$, given by the equation $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$, the following hold:*

a. If $P = (x(P), y(P))$ is a torsion point of $E(\mathbb{Q})$, then $4x(P) \in \mathbb{Z}$, and $8y(P) \in \mathbb{Z}$.
b. If the curve is of the form $y^2 = x^3 + Ax + B$, (i.e. if $a_1 = a_2 = a_3 = 0$) then $x(P) \in \mathbb{Z}$, $y(P) \in \mathbb{Z}$, and either $y(P) = 0$ or $y(P)^2$ divides $-4A^3 - 27B^2$.([4], p. 130)

To compute torsion subgroups using this theorem, we first change coordinates to put the equation of the curve into the form given in part (b), compute the quantity $\Delta = -4A^3 - 27B^2$, and, for $y = 0$ and each square divisor y^2 of Δ, solve the cubic equation $x^3 + Ax + (B - y^2) = 0$. For any integral solutions x of this equation, we test whether the point (x, y) is torsion (knowing by Theorem 1 that if it is torsion it has order at most 12). This algorithm is very straightforward, but has two major disadvantages–first, it requires that we factor Δ, which may be large and difficult to factor; second, even if Δ is easy to factor, it will often have many square divisors, and we must solve a cubic equation for each of them. We will see that this algorithm is rather slow, even for moderately sized Δ.

3. Bounding the size of the torsion subgroup

For an elliptic curve given by an equation of the form $y^2 = 4x^3 + Ax + B$, we may consider the set $E(\mathbb{F}_p)$ of points on the curve with coordinates in $\mathbb{F}_p$. For such points, we have the following result:

Theorem 3. *If p is a prime such that $p \nmid \Delta$, then there is a homomorphism from $E(\mathbb{Q})$ to $E(\mathbb{F}_p)$. If p is odd, then the restriction to the torsion subgroup of $E(\mathbb{Q})$ of the reduction homomorphism is injective.* ([5], p. 176.)

By applying theorem 3 with several small primes which do not divide Δ, we may quickly get a reasonably effective bound on the size of the torsion subgroup. Getting this bound is quite rapid–it involves a bounded number of computations, after reducing the coordinates of the curve mod p.

4. Analytic parameterization of elliptic curves

Given any lattice $\Lambda \in \mathbb{C}$, we define the Weierstrass $\wp$ function

$$\wp(z) = \frac{1}{z^2} + \sum_{\substack{\alpha \in \Lambda \\ \alpha \neq 0}} \left(\frac{1}{(z-\alpha)^2} - \frac{1}{\alpha^2} \right).$$

This function is doubly periodic, with period lattice Λ, and satisfies the differential equation

$$\wp'(z)^2 = 4\wp(z)^3 - g_2(\Lambda)\wp(z) - g_3(\Lambda),$$

where $g_2(\Lambda)$, and $g_3(\Lambda)$ are complex numbers depending on Λ. In addition, given any A and B, a lattice Λ exists with $-g_2(\Lambda) = A$, and $-g_3(\Lambda) = B$. Thus, if E is the elliptic curve given by $y^2 = 4x^3 + Ax + B$, we get a map $\mathbb{C}/\Lambda \to E(\mathbb{C})$ given explicitly by $z \mapsto (\wp(z), \wp'(z))$. This map is in fact an isomorphism of groups. This reduces the problem of finding rational torsion points of E to that of finding complex torsion points of $\mathbb{C}/\Lambda$, and checking whether their image in E is rational. However, the n-torsion points of $\mathbb{C}/\Lambda$ are trivially just $\frac{1}{n}\Lambda/\Lambda$, so we only need to check rationality of the images.

It is easy to see that the set of real points on an elliptic curve with real coefficients has either one or two components (in the standard topology on $\mathbb{R}^2$). The component which is unbounded will be called the identity component (since it can be considered to contain the point at infinity, which is the identity for the group law on the elliptic curve). There is a fast algorithm to calculate a basis of a lattice Λ of an elliptic curve with real coefficients using the AGM ([2], p. 391); the basis ω_1, ω_2 thus obtained has the property that the multiples of ω_1 correspond to points on the identity component of the curve, and the translates of these multiples by $\omega_2/2$ correspond to points on the bounded component (if any).

We note that computation of ω_1 and ω_2 is extremely rapid, since it requires only finding the roots of a cubic, and applying the AGM, which converges quadratically. Since part of the algorithm which we will present involves summing a series which converges linearly, the computation of ω_1 and ω_2 will not affect the asymptotic running time.

5. An analytic algorithm

Given an elliptic curve over $\mathbb{Q}$ (which we will assume is given by an equation $y^2 = 4x^3 + Ax + B$), we first calculate the number of points on the curve mod p for several small primes p which do not divide the discriminant of the curve. Since the torsion subgroup of $E(\mathbb{Q})$ injects into each of these groups, its size must divide each of the orders. Thus, we get a bound b on the order of the torsion subgroup. In fact the order of the torsion subgroup divides b.

If $b = 1$, then the torsion subgroup is trivial. We output this fact, and finish.

If $4 \not| b$, then by Mazur's Theorem we know that the torsion subgroup is cyclic of order at most 10. For each $n \leq 10$ and dividing b (beginning with the largest possible) we calculate a torsion point of order n on the identity component (corresponding to ω_1/n), and check whether it is a rational torsion point. If it is then it is a generator of the torsion subgroup $E(\mathbb{Q})$, and we output this fact and finish the algorithm. If it is not, and its order is even, it may differ from a rational torsion point by a two torsion point on the non-identity component, so we calculate two more torsion points (corresponding to $\omega_1/n + \omega_2/2$ and $\omega_1/n + \omega_1/2 + \omega_2/2$) and check whether either is rational. If one is, then it is a generator, and we output it and finish. If we complete this step without finding any rational torsion points, then the torsion subgroup is trivial; we output this fact and finish.

In the case where $4|b$, the torsion subgroup may not be cyclic. We calculate the two-torsion of the elliptic curve over $\mathbb{C}$ and check to see whether the points are rational.

If there is no two-torsion, then the torsion subgroup is cyclic of odd order, and must be entirely contained within the identity component of the real curve. We go through the odd divisors n of b less than or equal to 11 in descending order, and check whether ω_1/n corresponds to a rational torsion point. The first one which does is a generator; we output it and finish. If none of these points are rational, the torsion subgroup is trivial: we output this fact and finish.

If there is a single two torsion point, then the torsion subgroup is cyclic and we proceed to calculate it as in the case where $4 \not| b$, except that we need only check points of even order.

If there is more than one two torsion point, then we know that the torsion subgroup is non-cyclic. We may take a two torsion point on the non-identity component as one generator, and a point on the identity component of order 8,6,4 or 2, as the other generator. We compute the points corresponding to $\omega_1/8$, $\omega_1/6$ and $\omega_1/4$ for rationality. The first one which is rational is the second generator. If none is rational, then the second generator is the two torsion point on the identity component (which we have previously calculated).

6. Time analysis

To analyze the running time of the algorithm, we first need to know the size of the points with which we must work. We will assume that the curve is given in the form $y^2 = 4x^3 + Ax + B$. By the Lutz–Nagell theorem and the isomorphism of this curve with the curve $y^2 = x^3 + 4Ax + 16B$, we see that the y coordinate of a torsion point must satisfy $y^2 \leq |16^2(A^3 + 27B^2)|$. Further, the x-coordinate of a torsion point will be a root of $4x^3 + Ax + (B - y^2)$, and will thus have absolute value at most $\max(|A|, |B - y^2|) \leq \max(|A|, |B| + 16^2|A^3 + 27B^2|)$. Thus, in computing torsion, we only need to consider points with coordinates which are $O(\max(|A^3|, |B^2|))$ in magnitude. Let $C = 2^{13}\max(|A^3|, |B^2|)$. Arithmetic computations with numbers of magnitude $O(C)$ require $O(\log^2 C)$ bit operations. In computing with points on an elliptic curve, a bounded number of arithmetic operations are needed for each group operation on the curve, and in the entire algorithm, a bounded number of group operations are needed. Thus, the time to perform group operations in the algorithm is $O(\log^2 C)$.

The majority of the algorithm's running time is spent actually computing the points on the curve using the $\wp$ function. We note the following formula for $\wp(z)$ ([2], p. 389):

$$\wp(z) = \left(\frac{2\pi i}{\omega_1}\right)^2 \left(\frac{1}{12} + \frac{u}{(1-u)^2} + \sum_{n=1}^{\infty} q^n \left(u\left(\frac{1}{(1-q^n u)^2} + \frac{1}{(q^n - u)^2}\right) - \frac{2}{(1-q^n)^2}\right)\right),$$

where ω_1, ω_2 is a basis for the lattice Λ, $\tau = \omega_2/\omega_1$, $q = e^{2\pi i\tau}$, $u = e^{2\pi i z/\omega_1}$. In fact, by adjusting our basis for Λ, we may choose τ to have imaginary part at least $\sqrt{3}/2$, so that $|q| \leq k$, where $k = e^{-2\pi\sqrt{3}/2} = 4.33 \times 10^{-3}$. Also, by choosing z in a fundamental parallelogram for ω_1, ω_2, we guarantee that $|q| < |u| \leq 1$. Note that for the basis ω_1, ω_2 found by the AGM algorithm, the adjustment of the basis is extremely easy, requiring at most that we add one of the basis elements to the other.

We only need to compute torsion points to the nearest quarter integer (since if (x, y) is a torsion point on $y^2 = 4x^3 + Ax + B$, $(4x, 4y)$ is a torsion point on $y^2 = x^3 + 4Ax + B$, hence is integral), so we must add enough terms of the series that the remainder is less than 1/8. We note that each of the fractions inside the sum is bounded by a constant times $1/q^2$, so that after adding M terms of the sum, the error is less than $K_1 q^{M-2}$, where K_1 is a constant. This error is multiplied by $(2\pi/\omega_1)^2$, so we see that to get a good estimate of the number of terms we need, we must bound ω_1 away from 0.

From [1], p. 69, we find that

$$\omega_1^3 = \frac{2\pi^3}{\Delta^{1/4}}\theta_1^2(0)\theta_2^2(0)\theta_3^2(0)$$

where $\theta_1(0) = 2(q^{1/8} + q^{9/8} + q^{25/8} + ...)$, $\theta_2(0) = 1 - 2q^{1/2} + 2q^{4/2} - 2q^{9/2} + ...$, $\theta_3(0) = 1 + 2q^{1/2} + 2q^{4/2} + 2q^{9/2} + \dots$, and $\Delta = -A^3 - 27B^3$. It is easy to see that for our value of q, $|\theta_2(0)| > 1/2$, $|\theta_3(0)| > 1/2$, and $|\theta_1(0)| > |q|^{1/8}$. Thus,

$$|\omega_1| > \frac{\pi}{2|\Delta|^{1/12}}|q|^{1/12}.$$

In order to guarantee that the error,

$$\left(\frac{2\pi}{\omega_1}\right)^2 K_1|q|^{M-2} < \frac{4\pi^2|\Delta|^{1/6}K_1q^{M-2-1/6}}{(\pi/2)^2} \leq 16K_1|\Delta|^{1/6}k^{M-13/6}$$

is less than 1/8, we see easily that we need $O(\log|\Delta|) = O(\log C)$ terms. We compute each term to approximately $2 + \log_{10} C$ decimal places, since we know that the rational torsion points will have x-coordinates of size less than C. If our computation gives a point with coordinates which are too large, it is ignored, as it can not be a torsion point. On the other hand, if it is of the appropriate size, then the precision specified is adequate to identify it to the nearest quarter integer. Since we see that we need to compute $O(\log C)$ terms, using arithmetic operations requiring time $O(\log^2 C)$, the complexity of computing the Weierstrass $\wp$ function is $O(\log^3 C)$. A similar analysis with the same result holds for the $\wp'$ function, and, since we need to compute a bounded number of points, and all other calculations in the algorithm have a lower complexity, the running time of the algorithm is $O(\log^3 C)$.

7. GP-PARI code for the algorithm

The analytic algorithm for computing torsion has been coded in GP-PARI[1], and is available at http://www.math.uiuc.edu/~doud/torsion.html. Comparison with other computer algebra systems, which implement the Lutz–Nagell algorithm, shows that the analytic algorithm is not only faster for curves with extremely large coefficients, but also for many curves with small coefficients. We compare the analytic algorithm implemented in GP-PARI with the following torsion routines which use versions of the Lutz–Nagell algorithm: The C program torsion implemented by J. Cremona and following the algorithm in [3], the Maple program APECS (available by anonymous FTP at math.mcgill.ca) implemented by I. Connell, and the commercial computer algebra system Magma2 (developed by J. Cannon, *et. al.*). Table 1 gives the times in seconds required to compute the torsion subgroups of the following curves on a Sparc 5 workstation:

[1] The algorithm has also been implemented in C++ by John Cremona.

$$E_1 : y^2 = x^3 - 587044371375x + 173122882731933750,$$
$$E_2 : y^2 = x^3 + 337x^2 + 20736x,$$
$$E_3 : y^2 = x^3 + 4715281x^2 + 2520473760000x.$$

The first curve has torsion subgroup isomorphic to $\mathbb{Z}/3\mathbb{Z}$, and the other two have torsion subgroup isomorphic to $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/8\mathbb{Z}$.

Table 1. Running times in seconds

Curve	Analytic	Cremona	APECS	Magma
E_1	.225	9.1	1.2	5.53
E_2	.110	1.15	0	.416
E_3	.270	118	10.7	48.4

Note that the second curve is contained in the catalog of curves for which APECS has precomputed torsion, hence APECS does not compute the torsion–it looks it up. Also, the table does not include results for the torsion routine built into GP-PARI, since it was unable to calculate the torsion for curves E_1 and E_3 (it needed more stack space). For curve E_2 it took 23.6 seconds to calculate the torsion.

References

1. Chandrasekharan, K.: Elliptic Functions, Berlin–Heidelberg–New York: Springer 1985
2. Cohen, H.: A Course in Computational Algebraic Number Theory, Berlin–Heidelberg–New York: Springer, 1993
3. Cremona, J.: Algorithms for Modular Elliptic Curves, Cambridge: Cambridge University Press, 1992
4. Knapp, A.: Elliptic Curves, Princeton: Princeton University Press, 1992
5. Silverman, J.: The Arithmetic of Elliptic Curves, Berlin–Heidelberg–New York: Springer, 1986

This article was processed by the author using the TEX style file *Pjour1g* from Springer-Verlag