New sums of three cubes

Andreas-Stephan Elsenhans and Jörg Jahnel

Universität Göttingen, Mathematisches Institut, Bunsenstraße 3–5,
D-37073 Göttingen, Germany*
`elsenhan@wwwuser.gwdg.de`, `~jjahnel@wwwuser.gwdg.de`

Introduction. It is a long standing problem whether every rational integer $n \not\equiv 4, 5 \pmod 9$ may be written as a sum of three integral cubes. According to the web page `http://cr.yp.to/threecubes.html` of Daniel Bernstein, the first attacks by computer were carried out as early as in 1955.

Nevertheless, for example for $n = 3$, there is still no solution known different from the obvious $(1, 1, 1)$, $(4, 4, -5)$, $(4, -5, 4)$, and $(-5, 4, 4)$. For $n = 30$, the first solution was found by N. Elkies and his coworkers in 2000 [El]. It is interesting to note that, in 1992, D. R. Heath-Brown [HB] had made some prevision on the density of the solutions for $n = 30$ without knowing any solution, explicitly.

During the years, a number of algorithms have been developed in order to attack that problem. The historically first one which has a complexity of $O(N^{1+\varepsilon})$ for a search bound of N is the method of R. Heath-Brown [HLR]. The best algorithm presently known is Elkies' method described in [El].

Elkies' method. This is an algorithm which is geometric in nature. The idea is to cover the curve $Y = \sqrt[3]{1 - X^3}$, $X \in [0, 1/\sqrt[3]{2}]$, by very small parallelograms which we call *flagstones*. The algorithm finds all rational points of the particular form $(x/z, y/z)$ which are contained in one of the flagstones for $z \in \mathbb{N}$ up to a given bound N.

For each flagstone, this is equivalent to the detection of all points of the standard lattice $\mathbb{Z}^3$ which are contained in a certain pyramid. The problem is that, viewed in the standard coordinates, this pyramid has an enormous height in comparison with the two other dimensions. So to say, it has an extremely sharp apex. Searching naively for lattice points in such a pyramid would be highly inefficient.

The idea to overcome this difficulty is to work in coordinates more adapted to this pyramid. Then, the drawback is that the base $\{ (1, 0, 0), (0, 1, 0), (0, 0, 1) \}$ of the standard lattice gets far from being reduced in whatever sense. One needs to apply lattice basis reduction. Having done that, searching for lattice points within the pyramid is essentially equivalent to a search for small points of the lattice. For that, one may use the well-known algorithm of Fincke-Pohst [FP].

* The computer part of this work was executed on the Sun Fire V20z Servers of the Gauß Laboratory for Scientific Computing at the Göttingen Mathematisches Institut. Both authors are grateful to Prof. Y. Tschinkel for the permission to use these machines as well as to the system administrators for their support.

The size of the flagstones is somewhat arbitrary. Smaller flagstones mean that more time is required for lattice basis reductions. Larger ones lead to more time spent on the algorithm of Fincke-Pohst. The optimum depends on details of the implementation.

Implementation. Our implementation of Elkies' method is written in `C` and `C++`. We took care that only initialization parts of the code were written in `C++` or made use of the multi precision floats of `GMP`.

The time-critical parts were written in plain `C` making some use of the instruction `asm`. It turned out that, for most of the computations, 128-bit fixed-point arithmetic was sufficiently precise. We realized the 128-bit fixed-point numbers as arrays consisting of two `long int`s. The arithmetic of the fixed-point numbers was implemented in such a way that all loops (of length two) were manually unrolled.

For lattice basis reduction, we implemented a version of `LLL` for three-dimensional lattices. It turned out that adjacent flagstones lead to similar reduced bases. That is why an enormous optimization could be achieved by doing `LLL` *incrementally*. We start the `LLL` computation for the next flagstone with a reduced basis of the previous one and not with a naive basis.

Another substantial improvement was realized in the Fincke-Pohst part. Here, one has to compute many adjacent values of the same cubic polynomial in three variables. An implementation of a difference scheme reduced most of these computations to a few additions of values obtained before.

Some details. We searched systematically for solutions of $x^3 + y^3 + z^3 = n$ where the positive integer $n < 1000$ is neither a cube nor twice a cube and $|x|, |y|, |z| < 10^{14}$. The length of the flagstones was chosen dynamically. It was around $8.4 \cdot 10^{-12}$ near $x = 0$ and around $6.6 \cdot 10^{-14}$ near $x = 1/\sqrt[3]{2}$. The area of the flagstones was essentially constant at a value near $1.7 \cdot 10^{-40}$. This led to a total number of a bit more than 10^{13} flagstones to deal with.

We chose the widths of the flagstones such that all points in a horizontal distance of $<10^{-30}$ from the curve are contained in one of the flagstones. This should make sure that all solutions of heights between 10^{11} and 10^{14} are certainly found. Indeed, if we arrange variables such that $|x| \leq |y| \leq |z|$ then the point $(|x/z|, |y/z|)$ is in a horizontal distance from the curve of, in first order approximation, $\frac{ds^{1/3}}{ds}|_{s=(1-X^3)} \cdot n/|z|^3$. The derivative is always less than 0.53 since $X := |x/z| < 1/\sqrt[3]{2}$.

The whole search took around ten months of CPU time. Only 14% of that time was spent on lattice basis reductions. The lion's share was spent searching for small lattice points, i.e. on our implementation of the algorithm of Fincke-Pohst.

Results. In comparison with the list, dating back to 2001 and published on `http://cr.yp.to/threecubes.html`, 3 520 new solutions have been found.

Among them, there are solutions for $n = 52$, 156, 318, 366, 420, 564, 758, 789, 894, and 948. For each of these numbers, no solution was known in 2001.

For example, our computations show

$$\begin{aligned} 52 &= 60\,702\,901\,317^3 + 23\,961\,292\,454^3 - 61\,922\,712\,865^3 \\ &= 1\,232\,911\,859\,663^3 + 343\,101\,441\,461^3 - 1\,241\,705\,896\,626^3. \end{aligned}$$

For 13 values of n, for which exactly one solution was known in 2001, we found a second one. Among those, there is $n = 30$. The second solution for $n = 30$ looks like this,

$$30 = 3\,982\,933\,876\,681^3 - 636\,600\,549\,515^3 - 3\,977\,505\,554\,546^3.$$

A second and a third solution for $n = 75$ are as follows,

$$\begin{aligned} 75 &= 2\,576\,191\,140\,760^3 + 1\,217\,343\,443\,218^3 - 2\,663\,786\,047\,493^3 \\ &= 59\,897\,299\,698\,355^3 - 47\,258\,398\,396\,091^3 - 47\,819\,328\,945\,509^3. \end{aligned}$$

A complete list of all 14 288 solutions we know for $n < 1\,000$, n being neither a cube nor twice a cube, is available from the second author's web page `http://www.uni-math.gwdg.de/jahnel` as the file `threecubes_20070419.txt`. It was formed merging together the new solutions, D. Bernstein's lists from 1999 and 2001, and a list of small solutions found by a naive search.

Unfortunately, we still do not know any solution for $n = 33$ or $n = 42$. To say it more generally, the question whether $x^3 + y^3 + z^3 = n$ has an integral solution remains open for 14 numbers below 1 000. These numbers are 33, 42, 74, 114, 165, 390, 579, 627, 633, 732, 795, 906, 921, and 975.

References

[El] Elkies, N. D.: *Rational points near curves and small nonzero* $|x^3 - y^2|$ *via lattice reduction,* in: Algorithmic number theory (Leiden 2000), Lecture Notes in Computer Science 1838, Springer, Berlin 2000, 33–63

[FP] Fincke, U. and Pohst, M.: *Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,* Math. Comp. 44 (1985), 463–471

[HB] Heath-Brown, D. R.: *The density of zeros of forms for which weak approximation fails,* Math. Comp. **59** (1992), 613–623

[HLR] Heath-Brown, D. R., Lioen, W. M., and te Riele, H. J. J.: *On solving the diophantine equation* $x^3 + y^3 + z^3 = k$ *on a vector computer,* Math. Comp. 61 (1993), 235–244