THE SMALLEST SOLUTIONS
TO THE DIOPHANTINE EQUATION
$$x^6 + y^6 = a^6 + b^6 + c^6 + d^6 + e^6$$

GIOVANNI RESTA AND JEAN-CHARLES MEYRIGNAC

ABSTRACT. In this paper we discuss a method used to find the smallest non-trivial positive integer solutions to $a_1^6 + a_2^6 = b_1^6 + b_2^6 + b_3^6 + b_4^6 + b_5^6$. The method, which is an improvement over a simple brute force approach, can be applied to search the solution to similar equations involving sixth, eighth and tenth powers.

1. INTRODUCTION

Diophantine equations of the form $\sum_{i=1}^{m} a^s = \sum_{j=1}^{n} b^s$ have attracted interest since antiquity, and solutions for various values of s, m and n are known. The most comprehensive report to date is [1], while the first systematic computer search is described in [2]. In this paper we determine the first nontrivial solutions to the equation

$$(1) \qquad a_1^6 + a_2^6 = b_1^6 + b_2^6 + b_3^6 + b_4^6 + b_5^6.$$

The first 5 primitive solutions, i.e., with $\gcd(a_1, a_2, b_1, \ldots, b_5) = 1$, are

$$
\begin{aligned}
1117^6 + 770^6 &= 1092^6 + 861^6 + 602^6 + 212^6 + 84^6 \\
2041^6 + 691^6 &= 1893^6 + 1468^6 + 1407^6 + 1302^6 + 1246^6 \\
2441^6 + 752^6 &= 2184^6 + 2096^6 + 1484^6 + 1266^6 + 1239^6 \\
2827^6 + 151^6 &= 2653^6 + 2296^6 + 1488^6 + 1281^6 + 390^6 \\
2959^6 + 2470^6 &= 2954^6 + 2481^6 + 850^6 + 798^6 + 420^6.
\end{aligned}
$$

In the following we discuss the method used to find the solutions, which is based on a brute force decomposition algorithm made feasible by restricting the trials by means of modular arithmetic considerations.

2. THE METHOD

Since $x^6 \equiv 0 \pmod 7$ (resp., $x^6 \equiv 1 \pmod 7$) when 7 divides (resp., does not divide) x, for any $i = 1, 2, \ldots, 6$ we have $x_1^6 + \cdots + x_i^6 \equiv n_x \pmod 7$, where n_x denotes the number of x_i which are not divisible by 7.

Thus in any primitive solution to (1) we have either $n_a = n_b = 1$ or $n_a = n_b = 2$.

Either way, we can assume, reordering the b_i's if necessary, that 7 divides b_3, b_4 and b_5. Hence we have

$$(2) \qquad\qquad b_2^6 \equiv a_1^6 + a_2^6 - b_1^6 \pmod{7^6}.$$

Since we are looking for primitive solutions, we have assumed that $n_a = n_b > 0$. Hence at least one of b_1 and b_2 must be not divisible by 7. Thus we can assume that b_2 (and hence $a_1^6 + a_2^6 - b_1^6$) is not divisible by 7.

Our method, which is inspired by the one used in [2] to search (without success) for solutions to $a_1^6 = b_1^6 + b_2^6 + b_3^6 + b_4^6 + b_5^6 + b_6^6$, is based on the above considerations, and proceeds as follows.

For every a_1, $a_2 \le a_1$ and b_1 such that $a_1^6 + a_2^6 \ge b_1^6$ and $a_1^6 + a_2^6 - b_1^6 \not\equiv 0 \pmod{7^6}$, we solve the congruence

$$(3) \qquad\qquad x^6 \equiv a_1^6 + a_2^6 - b_1^6 \pmod{7^6}$$

with respect to x, obtaining a suitable set B of possible values for b_2.

The modular equation $x^6 \equiv t \pmod{7^6}$ has exactly 6 solutions below 7^6 when $t \equiv 1 \pmod 7$, and no solutions at all for other values of t which are non-zero mod 7. Hence the set B has a cardinality that does not exceed $6\lceil (a_1 + a_2)/7^6 \rceil$, and can be easily determined, once the solutions to $x^6 \equiv t \pmod{7^6}$ for $0 < t < 7^6$ have been precomputed.

For every value $b_2 \in B$ such that $v = a_1^6 + a_2^6 - (b_1^6 + b_2^6)$ is positive, we try to decompose $v/7^6$ as $v/7^6 = c_1^6 + c_2^6 + c_3^6$. If such c_j's do exist, we have obtained a solution to (1), where $b_3 = 7\,c_1$, $b_4 = 7\,c_2$, and $b_5 = 7\,c_3$.

This method, despite its simplicity, is faster than the naïve brute force approach based on the trial decomposition of $a_1^6 + a_2^6$ as a sum of five sixth powers. This depends mainly on two factors: a) the few possible values for b_2 (at most 6 in the range we have so far explored, i.e., $a_1, a_2 \le$ 30,400), and b) the trial decomposition of $v/7^6$ instead of v.

The method can be further refined by noticing that the property of the sixth powers that we have described for modulus 7 also holds with respect to moduli 8 and 9. In other words, $x^6 \pmod 8$ is 0 for x even and 1 for x odd, and analogously $x^6 \pmod 9$ is 0 for x a multiple of 3 and 1 otherwise.

The first immediate consequence is that we can discard all the configurations such that a_1 and a_2 are both multiples of 2 or 3, since we cannot obtain a primitive solution. More generally, we can exploit the following two observations to reduce the efforts needed to decompose a number y into the sum $x_1^6 + \cdots + x_k^6$ for $k < 7$.

- If $y \equiv 0 \pmod 7$ ((mod 8) or (mod 9), respectively), then 7 (2 or 3, respectively) must divide all the x_i's. In turn this implies that the problem can be reduced to the decomposition of $y/7^6$ (respectively $y/2^6$ or $y/3^6$) if such a number is an integer. If not, the decomposition does not exist.
- If $y \equiv 1 \pmod 7$ ((mod 8) or (mod 9), respectively), then we can assume that one of the x_i's, say x_1, satisfies $y^6 \equiv x_1^6 \pmod{7^6}$ ((mod 2^6) or (mod 3^6), respectively) and we can restrict the search space accordingly.

Moreover, all the values $x^6 > 0$ can be expressed as $(8k + 1)2^n$ for suitable values of k and n. This implies that $x^6 + y^6$, for every $x, y > 0$, is of the form $(4j + 1)2^m$. Therefore it is possible to devise a simple criterion, based on the binary decomposition of a number t, to discard several values of t that cannot be expressed as $x^6 + y^6$.

Finally we observe that straightforward modular considerations allow us to further restrict the search space. For example, we note that the sum of 3 sixth powers cannot be equal to 4 or 9 (mod 13), nor equal to 5, 16 or 17 (mod 19). Similar constraints, with respect to moduli 13, 19, 31, 37, 43, 61, 67, 73, 79, 109, 121, 139, 223 and 529, have been used to discard illegal values during the decomposition.

3. IMPLEMENTATION

The method was first implemented in C, using a public domain package for multiple precision arithmetic and a custom 91-bits integer representation for the trial decomposition. The first version of the program did not use the technique explained above to speed up the trial decomposition of $a_1^6 + a_2^6 - b_1^6$.

The program used about 20Mb of memory, essentially to maintain a sorted array of all the values $x^6 + y^6$ under a suitable limit to improve the performance of the trial decomposition.

The program ran on a PC with a Pentium2 CPU at 400MHz and 256Mb of central memory, under Windows NT, and found the smallest solution in about 5 minutes.[1]

TABLE 1.

a_1	a_2	b_1	b_2	b_3	b_3	b_5
1117	770	1092	861	602	212	84
2041	691	1893	1468	1407	1302	1246
2441	752	2184	2096	1484	1266	1239
2827	151	2653	2296	1488	1281	390
2959	2470	2954	2481	850	798	420
6623	323	6615	2912	642	434	363
8905	347	8820	5489	2576	2499	534
8969	5203	9023	2604	2520	2379	478
9551	9451	10080	8589	4884	3976	3943
9612	7271	9198	7446	6580	6279	5118
9707	6277	9675	5796	5531	4536	3640
12272	2459	11522	9144	8283	7434	1400
13417	5933	11403	11378	10698	4641	70
13903	5317	13788	8295	6026	1827	1232
15149	6914	14520	10675	10444	6006	810
15627	1865	14196	13608	5385	4032	565
15905	10519	14679	13759	9534	3822	2482
16103	9111	14868	13755	8484	6253	2004
16159	10532	15442	11760	11016	10273	3234
16160	1865	15792	11470	4984	4329	1008
16315	8749	15246	11682	11375	11140	7119
16867	14786	17790	9821	9786	2912	2502
17737	16043	17646	14574	14261	3255	2590
19168	1747	18984	10584	10088	8214	5299
19869	10673	19866	9783	9324	2088	1925

---

[1] The smallest solution to the equation under investigation here has also been discovered, independently, by E. Brisse while he participated in the project [4].

Subsequently, we implemented a version which takes into account all the techniques illustrated in the previous section, and which is written in assembly language to further improve performance.

Table 1 summarizes the primitive solutions in the range $a_1, a_2 < 20,000$. The 5 smallest were found by means of the first implementation of the method, and the others were found by the computational efforts of the participants in the project [4], to whom we have granted access to the improved version of our algorithm.

4. FURTHER WORK

So far, we have computed all solutions with $a_1, a_2 \leq 30{,}400$. In particular, no solution to the related equation $a_1^6 + a_2^6 = b_1^6 + b_2^6 + b_3^6 + b_4^6$ exists below 30,400. The similar equation $a_1^6 = b_1^6 + b_2^6 + b_3^6 + b_4^6 + b_5^6 + b_6^6$ is the subject of a current distributed search [4]. So far no solution has been found for $a_1 < 730,000$.

Following the heuristic arguments of [3], one might expect that there are $O(x)$ solutions to (1), as the variables run in the range $0, \ldots, x$. Indeed there are $O(x^7)$ choices for the seven variables, while $|a_1^6 + a_2^6 - b_1^6 - b_2^6 - b_3^6 - b_4^6 - b_5^6|$ lies in the range $0, \ldots, 5x^6$. Assuming, quite forcefully, that each number in this range is taken with equal probability, then we expect that the number of solutions is proportional to x. Experimental results support this line of reasoning, since the total number of solutions with the variables in the ranges $0, \ldots, 10,000$, $10,000, \ldots, 20,000$ and $20,000, \ldots, 30,000$ is 27, 46 and 52, respectively, or 10, 15, and 18, counting only the primitive solutions.

Some algorithms based on the same ideas exploited in this paper are being used to search for solutions to similar equations involving sums of eighth and tenth powers. As an example, for eighth powers, we use reductions with respect to 256 and 5^8 instead of 8 and 7^6, and modular constraints 17, 841 and 1681. For tenth powers, we use 2^{10} and 11^5 and moduli 25, 31, 41, and 61. Several results concerning eighth and tenth powers have been obtained, and an updated list of solutions is maintained in [4].

Scott I. Chase independently searched 8th, 10th, 12th and 16th powers with a similar method, and found several solutions, among which the most interesting is $966^8 + 539^8 + 81^8 = 954^8 + 725^8 + 481^8 + 310^8 + 158^8$.

ACKNOWLEDGMENTS

We wish to thank an anonymous referee for pointing out reference [3], and suggesting some extensions to our results.

REFERENCES

1. R.L. Ekl, *New Results in Equal Sums of Like Powers.* Math. Comp. **67** (1998), 1309–1315. MR **98m:**11023
2. L.J. Lander, T.R. Parkin, J.L. Selfridge, *A Survey of Equal Sums of Like Powers.* Math. Comp. **21** (1967), 446–459. MR **36:**5060
3. B. Mazur, *Questions about Powers of Numbers.* Notices of the AMS, February 2000 195–202. MR **2000k:**11111
4. J.-C. Meyrignac, et al., *Computing Minimal Equal Sums of Like Powers.* Distributed computing project, see `http://euler.free.fr`.

ISTITUTO DI MATEMATICA COMPUTAZIONALE -CNR, PISA, ITALY.
E-mail address: `resta@imc.pi.cnr.it`

E-mail address: `euler@free.fr`