

NUMERICAL RESULTS FOR SUMS OF FIVE AND SEVEN BIQUADRATES AND CONSEQUENCES FOR SUMS OF 19 BIQUADRATES

J.-M. DESHOUILERS AND F. DRESS

Dedicated to the memory of D. H. Lehmer

ABSTRACT. We describe the algorithms which allowed us to show that all the integers congruent to 50 modulo 80 that lie in the interval $(0.3651 \times 10^{12}, 1.0400 \times 10^{12})$ are sums of five biquadrates, and that all the integers congruent to 67 modulo 80 that lie in the interval $(0.3651 \times 10^{12}, 9.5956 \times 10^{18})$ are sums of seven biquadrates. We finally describe some ascent lemmas that we use to deduce from the previous results that every integer not exceeding 10^{448} is a sum of 19 biquadrates.

1. INTRODUCTION

In the first edition of his *Meditationes Algebraicae* in 1770, Waring states that every natural integer is a sum of 19 biquadrates (fourth powers of natural numbers). It seems obvious that he had no proof of this fact and that he merely observed that every small integer is the sum of 19 biquadrates, some of them (79 or 479, for example) requiring as much as 19 summands. The actual limit of his computations is not known.

Hardy and Littlewood [8] opened the road to the solution of Waring's problem by showing in 1925 that there exists an effectively computable constant A such that every integer larger than A is a sum of 19 biquadrates. In 1986, Balasubramanian, Deshouillers, and Dress showed that 10^{367} is an admissible value for A ; the result is announced in [1, 2] and the complete proof is published in [5–7].

The same notes [1, 2] contain a sketch of the computations performed to show that every integer smaller than 10^{378} is a sum of 19 biquadrates.

Following the same general line, we have redone these computations, and, thanks to the improvement in our computing facilities, we could increase the range of our results. In this article, we give a detailed account of how we proved the following

Theorem. *Every positive integer N less than 10^{448} is the sum of 19 biquadrates.*

Besides the result of [1], the best previous bound was $10^{233.5}$, obtained by

Received by the editor July 27, 1992.

1991 *Mathematics Subject Classification.* Primary 11P05, 11–04; Secondary 11B13, 11B57, 11Y16.

Thomas in 1974 (cf. [10] as well as [3] for a comment on that article).

2. GENERAL SCHEME OF THE PROOF

We shall use the notation B_s to describe an integer which is the sum of s biquadrates; thus “ m is B_s ” means that the integer m is the sum of s biquadrates.

The principle of the proof is traditional: we first show by a numerical verification that all (or almost all, with a *very* limited number of exceptions) integers in some finite arithmetical progressions are B_h ; we then deduce that all integers in a (much larger) interval are B_{19} by a $(19-h)$ -fold application of ascent arguments. The ascent (or descent) arguments involved are more or less refined versions of the greedy algorithm, which consists in looking for representations involving the largest possible biquadrate.

The numerical evidence strongly suggests that the proportion of B_5 's in any arithmetical progression $16n+1$, 2 , 3 , 4 , or 5 should tend to 1 when the length of the progression tends to infinity; theoretical considerations even support the conjecture that any sufficiently large integer in one of these progressions is a B_5 , but exceptions do exist rather far up (to 10^{15} ?). In order to apply the greedy algorithm, it is essential to be equipped with very long arithmetic progressions containing only B_5 's (with no exception at all, although dealing with very few exceptions is manageable, but costly, as we already showed in [1]).

In order to make a compromise between the length and the size of the needed arithmetic progression full of B_5 's, we restricted ourselves to an arithmetic progression of integers congruent to 50 modulo 80. Our choice is discussed in the third section. More precisely, we prove the following

Proposition 1. *Every integer lying in the interval*

$$[l_1, l_2] = (365\,093\,023\,490, 1\,040\,000\,000\,000)$$

which is congruent to 50 modulo 80 is a B_5 .

This result improves on our previous one in that it contains a much larger interval in which we prove that there is no exception; it improves on Thomas' in that respect and also in that it concerns B_5 's (and not B_6 's), leaving one more step available in the ascent process. A further improvement on [8] is that we jump from B_5 's to B_7 's by a single ascent with B_2 's instead of two ascents with B_1 's. The key ingredient is the following

Proposition 2. *There exists a sequence of B_2 integers (g_j) congruent to 17 modulo 80, which contains 17 and an element larger than $9.595669026 \times 10^{18}$, such that all consecutive differences $g_{j+1} - g_j$ are at most equal to $\delta = l_2 - l_1 = 674\,906\,976\,510$.*

The standard greedy algorithm readily leads from Propositions 1 and 2 to the following

Corollary. *Every integer lying in the interval*

$$[m_1, m_2] = (365\,093\,023\,490, 9.59566939 \times 10^{18})$$

and congruent to 17 modulo 80 is a B_7 .

One should notice that a standard two-step ascent would lead to $1.04 \dots \times 10^{18}$ instead of $9.59 \dots \times 10^{18}$.

In the last section, we show how it is possible to choose 12 ascent steps managing congruences modulo 16 and 5, in order to deduce from the above corollary that every integer between 10^{12} and 10^{448} which is not divisible by 16 is a B_{19} . It has long been known that every integer less than 10^{12} is a B_{19} , so that every integer at most equal to 10^{448} which is not divisible by 16 is a B_{19} . Since 16 is a biquadrate, and since the product of two biquadrates is a biquadrate, it is obvious that every integer at most equal to 10^{448} is a B_{19} , which is our main theorem.

3. HEURISTICS ON SUMS OF BIQUADRATES

Since we need arithmetic progressions which are rich in sums of biquadrates, we restrict our attention to those moduli for which biquadrates are badly distributed. Modulo 16, every biquadrate is congruent to 0 or 1, and the solutions of $x^4 \equiv m \pmod{2^\alpha}$ for $\alpha \geq 5$ are simply lifted from those of $x^4 \equiv m \pmod{16}$. For an odd prime p , there are $\frac{p-1}{4}$ nonzero biquadratic residues mod p when p is congruent to 1 modulo 4, and $\frac{p-1}{2}$ otherwise; the solutions of $x^4 \equiv m \pmod{p^\alpha}$ for $\alpha \geq 2$ are simply lifted from those of $x^4 \equiv m \pmod{p}$. These considerations, in combination with practical ones, which require that our modulus should not be too large, led us to choose $16 \times 5 = 80$.

It is easy to check that m^4 is congruent to 0 (resp. 1, 16, or 65) modulo 80 if and only if m is congruent to 0 (resp. ± 1 or ± 3), $(\pm 2$ or $\pm 4)$, or 5) modulo 10. We thus get Table 1.

TABLE 1

remainder modulo 80	frequency $\times 10$
1	4
16	4
0	1
65	1

Table 2 (next page) follows from Table 1, and gives the remainders modulo 80 which are B_2 , as well as the ordered representations (e.g., 17 can be realized as $1 + 16$ or $16 + 1$) and the frequency with which a B_2 is congruent to the given remainders modulo 80.

The highest frequency is attained for the remainder 17, and this is why we chose it in the B_2 ascent process (cf. Proposition 2).

With slightly more work, we can construct Table 3 (next page), associated with the B_5 's. It turns out from this table that the richest arithmetic progressions for our purpose are $80n + 19$ and $80n + 34$, as already noticed by Thomas.

However, in order to reduce the complexity of the program, that is, both the human work and the risk of error, we chose one of the richest representations according to Table 4 (see p. 199).

The choice between 35 and 50 was purely random (and it may turn out that 35 would have been a slightly better choice). A further advantage of either choice

TABLE 2

remainder modulo 80	ordered representations	frequency $\times 100$
17	1 + 16	32
1	0 + 1 16 + 65	16
2	1 + 1	16
32	16 + 16	16
16	0 + 16	8
66	1 + 65	8
65	0 + 65	2
0	0 + 0	1
50	65 + 65	1

TABLE 3

remainder modulo 80	ordered representations	frequency $\times 10^5$
19	0 + 1 + 1 + 1 + 16 1 + 1 + 16 + 16 + 65	12 800
34	0 + 1 + 1 + 16 + 16 1 + 16 + 16 + 16 + 65	12 800
35	0 + 0 + 65 + 65 + 65 1 + 1 + 1 + 16 + 16	10 250
50	0 + 0 + 0 + 65 + 65 1 + 1 + 16 + 16 + 16	10 250
3	0 + 0 + 1 + 1 + 1 0 + 1 + 1 + 16 + 65 1 + 16 + 16 + 65 + 65	6 400
4	0 + 1 + 1 + 1 + 1 1 + 1 + 1 + 16 + 65	6 400
18	0 + 0 + 1 + 1 + 16 0 + 1 + 16 + 16 + 65 16 + 16 + 16 + 65 + 65	6 400
49	0 + 1 + 16 + 16 + 16 16 + 16 + 16 + 16 + 65	6 400
20	0 + 65 + 65 + 65 + 65 1 + 1 + 1 + 1 + 16	5 125
65	0 + 0 + 0 + 0 + 65 1 + 16 + 16 + 16 + 16	5 125
...

is that it is then easy to check whether an integer which has no representation of the selected type is a B_5 or not: N is a B_5 if and only if N is divisible by $625 = 5^4$ and $N/625$ is itself a B_5 . Since N is at most 1.04×10^{12} , $N/625$ is at most 1.664×10^9 , and it is easy to check directly whether such a “small” number is a B_5 .

As we already mentioned, biquadrates are badly distributed in arithmetic progressions modulo 13, since 13 is a prime congruent to 1 modulo 4. This uneven distribution is reflected in that of the B_s 's for small s . For example, the asymptotic frequency of the B_5 's in progressions modulo 13 is given in Table 5. This will be referred to in §5.

TABLE 4

remainder modulo 80	one ordered representation	frequency $\times 10^5$
35	1 + 1 + 1 + 16 + 16	10 240
50	1 + 1 + 16 + 16 + 16	10 240
19	1 + 1 + 16 + 16 + 65	7 680
34	0 + 1 + 1 + 16 + 16	7 680
19	0 + 1 + 1 + 1 + 16	5 120
34	1 + 16 + 16 + 16 + 65	5 120
...

TABLE 5

remainder modulo 13	frequency of the B_5 's
0	0.0517
1	0.0707
2	0.0773
3	0.0707
4	0.1008
5	0.0773
6	0.0773
7	0.0672
8	0.0672
9	0.0707
10	0.1008
11	0.0672
12	0.1008

4. ALGORITHMS

4.1. **On the B_5 's congruent to 50 modulo 80.** Our aim is to show how we proved Proposition 1. As we explained in the previous section, we checked whether an integer $80n + 50$ with $n \leq 1.3 \times 10^{10}$ is a B_5 with summands congruent to 1, 1, 16, 16, 16 modulo 50. When this was not the case, we checked whether $80n + 50$ was divisible by 625. Since this *never* happened, there is no point in describing the treatment which would have been adequate.

We start by constructing all the sums a_2 of B_2 's which are congruent to 2 modulo 80, as well as the sums a_3 of B_3 's which are congruent to 48 modulo 80. Since we consider integers which are less than 1.04×10^{12} , either set has less than 1.1×10^7 elements, which is manageable. However, the file of the

sums $\{a_2\} + \{a_3\}$ would require about 1 500 megabytes, which is more than our available disk space.

To overcome this difficulty, we break the files $\{a_2\}$ and $\{a_3\}$ into arithmetic progressions to a suitable modulus M : distributing $\{a_2\}$ and $\{a_3\}$ in arithmetic progressions is straightforward and costs almost no time; in order to recover all the sums $a_2 + a_3$ in a given progression modulo M with remainder r , it is enough to consider the M pairs (r', r'') with $r' + r'' \equiv r \pmod{M}$ and for each such pair to sum the a_2 's congruent to r' modulo M and the a_3 's congruent to r'' modulo M . Let us consider the time and space costs in this process, under the assumption that M is a prime number congruent to 3 modulo 4, since in this case the sets $\{a_2\}$ and $\{a_3\}$ are rather well distributed modulo M :

- the total time is almost unchanged: we have to perform $M \times M$ times a sequence of operations, each of which needs about $1/M^2$ less time than the initial full sequence;
- on the other hand, the needed space in memory is simply divided by M .

We chose $M = 43$, and the needed memory was reduced to 38 megabytes. We should notice here that we indeed performed a further block reduction, so that the sequences we actually handled were stored on blocks of length at most eight megabytes, although we are not going to describe that minor aspect any further.

We worked with two algorithms:

1. Computation of the biquadrates, then of the sets $\{a_2\}$ and $\{a_3\}$; those sets are ordered modulo 43, and stored into 86 files: each of the first 43 contains the list of those integers k such that $80(43k + i) + 2$ is in $\{a_2\}$, for $0 \leq i \leq 42$, each of the remaining 43 files contains the list of those integers k such that $80(43k + i) + 48$ is in $\{a_3\}$, for $0 \leq i \leq 42$.

2. Search (between two given bounds) for the exceptional n 's such that $80n + 50$ is not in $\{a_2\} + \{a_3\}$, by means of two loops: the external one is indexed by the remainder r modulo 43 of n . The internal one is indexed by the pairs (r', r'') such that $r' + r'' \equiv r \pmod{43}$. Furthermore, if an integer $80n + 50$ is not recognized as a B_5 , it is checked whether $80n + 50$ is divisible by 625, in which case $(80n + 50)/625$ is declared as a "pre-exception" in order to be analyzed by ad hoc methods. (As we already mentioned, we do not describe these methods, since no large pre-exception was encountered.)

Programs were written in the C language, and run on a SUN workstation. We briefly describe the algorithms underlying those programs.

Algorithm 1a: construction of the 43 files for $\{a_2\}$.

procedure main

1. computation of biquadrates $n^4 \equiv 1 \pmod{80}$, decomposition $n^4 = 80(43k + r) + 1$, and storage of k and r in two arrays
2. computation through a double loop of the "k-values" of the sums of two elements in $\{a_2\}$, and immediate storage in the one of the first 43 files which is associated with the "r-value" of the sum
3. loop on the 43 files
 - 3.1. call of procedure sort
 - 3.2. rewriting of each of the 43 sorted files
4. end

procedure sort

writing a file of integers in increasing order, and elimination of multiple values

Algorithm 1b: construction of the 43 files for $\{a_3\}$. Similar to Algorithm 1a.

The running time of the respective programs was about one hour with our numerical values.

Algorithm 2: search for the exceptions $(\equiv 50 \pmod{80})$ and not B_5 .

procedure main

1. input the limits for the search, m_1 and m_2
2. reduction step: $n_i = [m_i/80]$, $k_i = [n_i/43]$ ($i = 1, 2$)
3. loop from $r = 0$ to $r = 42$
 - 3.1. store zeros in the array that will store the information (bit # j will be 1 (or 0) according as the checked integer is (or is not) a sum $a_2 + a_3$)
 - 3.2. loop from $r' = 0$ to $r' = 42$
call procedure test (r' , r'')
 - 3.3. detect pre-exceptions
 - 3.3.1. detect indices corresponding to pre-exceptions
 - 3.3.2. convert those indices into the values of the pre-exceptions
 - 3.4. detect exceptions and pre-exceptions
 - 3.4.1. if a pre-exception is not divisible by 625, post it as an exception
 - 3.4.2. otherwise, divide it by 625 and post it as a possible exception
4. end

procedure test (r_1, r_2)

1. read the file containing the “ k -values” for a_2 's with r -value r_1
2. read the file containing the “ k -values” for a_3 's with r -value r_2
3. double loop for writing sums $a_2 + a_3$ with r -value (r_1, r_2)
 - 3.1. computation of the “ k -value” of the sum $a_2 + a_3$
 - 3.2. store 1 in the corresponding bit
4. return

Running the respective program (written in C) on our SUN station took about 1 000 hours.

4.2. A sequence of B_2 's congruent to 17 modulo 80 with small differences. The third program is used to produce a sequence of B_2 's satisfying the conditions of Proposition 2. It was written in C and run on a SUN workstation. Its principle is fairly elementary: starting from $g_0 = 17$, a sequence of g_j 's is built by induction according to the following algorithm:

- for a given g_j , one looks for an integer m between g_j and $g_j + \delta$ which is a B_2 and congruent to 17 modulo 80; the greedy algorithm is used (searching for an m as close to $g_j + \delta$ as possible), which is very fast;
- in case of success, one lets g_{j+1} be equal to m , and one goes back to the beginning;

- otherwise, execution is stopped. (For $\delta = 0.674907 \times 10^{12}$, the last g_j found has the value $9.59566902 \times 10^{18}$ given in Proposition 2.)

A first modification is that we did not start with the value $\delta = 0.6749 \times 10^{12}$, but we determined the successive jumps of the function $y = f(x)$ defined by

$$f(x) = \max\{y \mid \sigma_j \leq y \Rightarrow \sigma_j - \sigma_{j-1} \leq x\},$$

where (σ_j) is the increasing sequence of B_2 numbers which are congruent to 17 mod 80. The function f is a nondecreasing step function characterized by the sequence (δ_k, y_k) defined by $f(x) = y_k$ for $\delta_k \leq x < \delta_{k+1}$.

The second modification reduces the computing time by a factor 10 (from 2000 hours to 200, which is crucial). Instead of computing the elements of the sequence (σ_j) one by one, we compute all the B_2 's in a given interval $[g_j, g_j + D]$ (where D is about 100δ to 1000δ), and instead of storing and sorting them, we just indicate their existence in an interval of length δ/t , by writing "1" in a box associated with the interval. These boxes are then looked at: if there is no consecutive block of $(t-2)$ boxes containing no "1", then we can go to the beginning of the algorithm, with $g_j + D\delta$ instead of g_j . Otherwise, a greedy search is used to determine g_{j+1} .

We briefly describe the structure of the algorithm underlying this last program.

Algorithm 3: generating a sequence of $B_2 \equiv 17 \pmod{80}$.

procedure main

1. input initial g and $delta$
2. infinite loop
 - 2.1. $boxflag = 1$
 - 2.2. loop while $boxflag = 1$
 - 2.2.1. all boxes are initialized with 0
 - 2.2.2. $lim = g + D \cdot delta$
 - 2.2.3. double loop on $2a$ not divisible by 5, and $2b + 1$ not divisible by 5, under the condition $g \leq s = (2a)^4 + (2b+1)^4 \leq lim$; the corresponding box is set to "1"
 - 2.2.4. search loop on the boxes: if there exists a block of $t - 2$ consecutive boxes containing only "0", then $boxflag = 0$
 - 2.3. continuation (in case $boxflag = 0$)
 - 2.4. $g = g + j(delta/t)$, where j is the index of the box preceding the block
 - 2.5. call procedure $greedysearch(g, delta)$

procedure greedysearch(g, delta)

1. double loop on $2a$ not divisible by 5, and $2b + 1$ not divisible by 5, under the condition $s = (2a)^4 + (2b + 1)^4 \leq g$
 - if one finds $s \leq g + delta$
 - end of the loop; $g = s$; end of the procedure
 - otherwise
 - $m = g + (delta/2)$; call procedure $followdelta(m)$; return

procedure followdelta(m)

1. exhaustive search to find the largest $mis = (2a)^4 + (2b + 1)^4 < m$
2. exhaustive search to find the smallest $mas = (2a)^4 + (2b + 1)^4 > m$

3. $\text{delta} = \text{mas} - \text{mis}; g = \text{mas}$
4. $\text{print mis, mas, delta}$

Coding this algorithm is straightforward.

5. NUMERICAL RESULTS

We present the last few values we found for numbers which are congruent to 50 mod 80, but not B_5 . Although our main theorem requires only the knowledge of a large block without exceptions, we provide those values for the following reasons:

- the first one is to provide data that can be checked if our computations are redone sometime;
- the second is to draw the reader's attention to the following facts:
 - * the large gap between the penultimate and the last observed exceptions,
 - * all 25 exceptions in the list, except 106 813 602 930 and 128 078 473 090, are divisible by 13 (a result to be compared with Table 5 in §3).

All the numbers congruent to 50 modulo 80 which lie in [100 000 000 000, 1 040 000 000 000] are B_5 , with the following exceptions:

			102 380 819 970
106 016 815 970	106 065 556 610	106 813 602 930	108 317 800 370
111 201 089 090	114 255 831 170	114 731 128 850	117 963 507 090
118 262 419 730	126 344 288 850	128 061 547 250	128 078 473 090
129 647 053 250	132 493 652 850	136 680 665 810	139 488 024 130
139 598 700 930	155 439 625 250	160 641 941 330	172 999 456 370
191 532 446 690	192 714 742 610	206 740 959 490	365 093 023 490

We provide the reader with an extended list of pairs (δ_k, γ_k) in Table 6. This table suggests that γ_k increases at a slightly slower rate than δ_k^2 .

6. THE DESCENT METHOD; END OF THE PROOF

Let N be an integer less than 10^{448} . In order to show that N is a B_{19} , it is enough to show that $N - ([N^{1/4}]^4)$ is a B_{18} , and the binomial expansion tells us that it is enough to know that any integer less than $4N^{3/4} = 10^{336.6}$ is a B_{18} . We see the power in the way the exponent is decreasing: with a few iterations, we see that it would be enough to show that any integer less than $10^{16.53}$ is a B_7 . This is, however, not true for two reasons: the first one is that B_7 's are located only in certain arithmetical progressions, and the second is that even when those arithmetical conditions are fulfilled, it is not possible to represent small integers as B_7 's. The second obstruction is easy to overcome by starting with a long arithmetical progression, like the one provided by the corollary in §2. The first one is a bit more cumbersome, and in order to master it, we have to control congruence conditions in the descent process.

6.1. The congruence descent. Starting with an integer N , we have to find twelve biquadrates b_1, b_2, \dots, b_{12} so that $N - b_1 - \dots - b_{12}$ is congruent to 67 modulo 80.

The first case deals with integers N congruent to 3, 4, 5, ..., 15 modulo 16. Since all biquadrates are congruent to 0 or 1 modulo 5 and to 0 or 1 modulo 16, there is no difficulty in finding such a sequence of b_i 's. It is even worth noticing that among four consecutive biquadrates, one at least is congruent to 1

TABLE 6. List of pairs (δ_k, γ_k) for γ_k between 10^{17} and 1.7×10^{19}

δ_k	γ_k
$0.077\,890\,2 \times 10^{12}$	$0.100\,407\,976 \times 10^{18}$
$0.081\,696\,8 \times 10^{12}$	$0.155\,422\,263 \times 10^{18}$
$0.083\,603\,6 \times 10^{12}$	$0.162\,580\,587 \times 10^{18}$
$0.083\,765\,9 \times 10^{12}$	$0.191\,777\,160 \times 10^{18}$
$0.088\,131\,8 \times 10^{12}$	$0.198\,809\,922 \times 10^{18}$
$0.090\,539\,5 \times 10^{12}$	$0.213\,490\,334 \times 10^{18}$
$0.090\,895\,9 \times 10^{12}$	$0.218\,150\,754 \times 10^{18}$
$0.099\,216\,6 \times 10^{12}$	$0.219\,814\,867 \times 10^{18}$
$0.101\,508\,8 \times 10^{12}$	$0.259\,162\,611 \times 10^{18}$
$0.106\,282\,8 \times 10^{12}$	$0.259\,842\,504 \times 10^{18}$
$0.110\,535\,4 \times 10^{12}$	$0.261\,322\,155 \times 10^{18}$
$0.111\,865\,3 \times 10^{12}$	$0.265\,137\,124 \times 10^{18}$
$0.113\,428\,4 \times 10^{12}$	$0.295\,537\,884 \times 10^{18}$
$0.118\,487\,6 \times 10^{12}$	$0.314\,854\,500 \times 10^{18}$
$0.151\,697\,2 \times 10^{12}$	$0.326\,498\,802 \times 10^{18}$
$0.155\,242\,6 \times 10^{12}$	$0.556\,745\,800 \times 10^{18}$
$0.155\,585\,0 \times 10^{12}$	$0.576\,318\,161 \times 10^{18}$
$0.175\,581\,4 \times 10^{12}$	$0.633\,794\,988 \times 10^{18}$
$0.186\,619\,3 \times 10^{12}$	$0.646\,110\,378 \times 10^{18}$
$0.198\,480\,2 \times 10^{12}$	$0.663\,736\,038 \times 10^{18}$
$0.202\,209\,4 \times 10^{12}$	$0.838\,097\,027 \times 10^{18}$
$0.246\,588\,5 \times 10^{12}$	$0.899\,789\,363 \times 10^{18}$
$0.350\,230\,4 \times 10^{12}$	$0.991\,660\,740 \times 10^{18}$
$0.351\,731\,2 \times 10^{12}$	$2.284\,134\,282 \times 10^{18}$
$0.380\,510\,2 \times 10^{12}$	$2.687\,993\,301 \times 10^{18}$
$0.394\,274\,7 \times 10^{12}$	$3.000\,022\,126 \times 10^{18}$
$0.437\,250\,2 \times 10^{12}$	$3.191\,452\,333 \times 10^{18}$
$0.441\,527\,2 \times 10^{12}$	$3.200\,070\,443 \times 10^{18}$
$0.442\,210\,1 \times 10^{12}$	$4.132\,544\,539 \times 10^{18}$
$0.445\,055\,0 \times 10^{12}$	$4.605\,906\,871 \times 10^{18}$
$0.470\,315\,9 \times 10^{12}$	$4.610\,024\,415 \times 10^{18}$
$0.550\,492\,8 \times 10^{12}$	$4.665\,433\,340 \times 10^{18}$
$0.613\,988\,6 \times 10^{12}$	$5.289\,622\,157 \times 10^{18}$
$0.633\,765\,5 \times 10^{12}$	$7.004\,516\,571 \times 10^{18}$
$0.651\,392\,3 \times 10^{12}$	$7.716\,762\,811 \times 10^{18}$
$0.660\,149\,7 \times 10^{12}$	$8.995\,568\,346 \times 10^{18}$
$0.668\,582\,7 \times 10^{12}$	$9.184\,529\,357 \times 10^{18}$
$0.683\,890\,4 \times 10^{12}$	$9.595\,669\,027 \times 10^{18}$
$0.712\,808\,3 \times 10^{12}$	$9.745\,398\,774 \times 10^{18}$
$0.834\,506\,7 \times 10^{12}$	$9.930\,335\,037 \times 10^{18}$
$0.857\,042\,7 \times 10^{12}$	$10.432\,599\,438 \times 10^{18}$
$0.868\,568\,1 \times 10^{12}$	$13.069\,930\,352 \times 10^{18}$
$0.870\,389\,8 \times 10^{12}$	$13.261\,088\,133 \times 10^{18}$
$0.912\,696\,8 \times 10^{12}$	$15.974\,961\,266 \times 10^{18}$
$0.963\,049\,9 \times 10^{12}$	$16.323\,537\,190 \times 10^{18}$
$1.076\,902\,9 \times 10^{12}$	$16.908\,631\,798 \times 10^{18}$
limit of the search:	$17.000\,000\,000 \times 10^{18}$

mod 5 and to 1 mod 16, and one at least is congruent to 1 mod 5 and to 0 mod 16: when b_i is chosen according to this principle, we say that we perform an i th step of S-type in the descent. In a similar way, steps of T-type refer to the fact that among ten consecutive biquadrates, one at least is congruent to 0 mod 5 and to 0 mod 16, and one at least is congruent to 0 mod 5 and to 1 mod 16.

One readily sees that it is always possible to go down from N congruent to 3, 4, 5, . . . , 15 modulo 16, to an integer congruent to 67 modulo 80, by using 12 steps, with at most four steps of T-type at the beginning, the remaining steps being of S-type. The worst cases concern integers N which are congruent to 15 and 35 modulo 80: there is no other choice modulo 16 than taking only biquadrates congruent to 1 in the first instance and 0 in the second. Modulo 5, we want to reduce the number of T-type steps; but four are needed, because $b_1 + \dots + b_{12}$ has to be congruent to 3 modulo 5 in both instances.

The second case deals with integers N congruent to 0 modulo 16; it turns out that this case can be completely ignored: it is enough to prove that all integers up to X which are not congruent to 0 modulo 16 are B_{19} , in order to show that all integers up to X are B_{19} : this is due to the fact that $16 = 2^4$ and that $16B_s$ is a B_s .

The third case, dealing with integers N congruent to 1 or 2 modulo 16, might seem hopeless if we attack it with the argument of the first case. Fortunately, after one or two steps of S-type (or even simpler), we encounter an integer congruent to 0 modulo 16, which we may divide by 16. We must however ensure that the quotient we get will entitle us to end the descent as in the first case: this will be achieved if we insure the quotient to be congruent to 3, 4, 5, . . . , 12 or 13 modulo 16, since only ten steps in the descent are still at our disposal if N is congruent to 2 modulo 16. The following lemma explains how we can deal with this step: its proof is just a direct computation.

Lemma 1. *Let m be an integer which is congruent to 1 modulo 16. Among any set of five consecutive odd integers, at least two of them (a_1 and a_2 , say) have the property that $(m - a_i^4)/16$ is an integer congruent to 3, 4, . . . , 12 or 13 modulo 16, for $i = 1$ and 2.*

A further advantage of this lemma is that it also permits us to manage congruence modulo 5 in this step: one of a_1^4 or a_2^4 is congruent to 1 modulo 5, so that the corresponding $(m - a_i^4)/16$ is congruent to $m - 1$ modulo 5. A descent step using this principle will be described as belonging to the U-type.

If N is congruent to 1 modulo 16, one can reduce it to an integer congruent to 67 modulo 80 by starting with a U-type step, followed by 11 steps consisting of at most four T-type steps preceding a sequence of S-type steps.

If N is congruent to 2 modulo 16, the corresponding sequence is an S- or a T-type step, followed by a U-type step, followed by ten steps consisting of at most three T-type steps preceding a sequence of S-type steps.

6.2. The real ascent. We give two lemmas which permit us to compute the length of an arithmetical progression containing B_{s+1} 's, when one knows an arithmetical progression containing only B_s 's.

Lemma 2. *Let A, s be positive integers, and D_s be a real number (larger than 40 000). If we assume that every integer in the interval $[A, A + D_s)$ which is congruent to r modulo 80 is a B_s , then every integer in the interval $[A, A + D_{s+1})$*

which is congruent to $r + \delta$ modulo 80 is a B_{s+1} , with the following values:

$$D_{s+1} = D_s + (D_s/c)^{4/3}$$

and

$$\begin{aligned} \delta = 1 \text{ or } 16, \quad c = 16 & \quad (\text{S-type ascent}); \\ \delta = 0 \text{ or } 65, \quad c = 40 & \quad (\text{T-type ascent}). \end{aligned}$$

Proof. This result is a slight variation of the standard greedy algorithm. Let h denote the number 4 (resp. 10) in the case of the S-type (resp. T-type) ascent. Among h consecutive integers, one of them at least has a fourth power which is congruent to δ modulo 80.

On the other hand, if we let $0 = a_0 < a_1 < \dots < a_k \leq (D_s/4h)^{1/3}$ be a sequence of integers such that $a_{i-1} - a_i \leq h$, the intervals $[A + a_{i-1}^4, A + D_s + a_{i-1}^4)$ corresponding to two consecutive values of i ($1 \leq i \leq k$) do overlap, since we have

$$\begin{aligned} A + D_s + a_{i-1}^4 &\geq A + D_s + (a_i - h)^4 \geq A + 4ha_i^3 + (a_i - h)^4 \\ &\geq A + a_i^4 + h^2(6a_i^2 - 4a_i h + h^2) \geq A + a_i^4. \end{aligned}$$

The lemma is a straightforward consequence of these two facts. \square

In a similar way, it is easy to deduce the next lemma from Lemma 1 and the comment that follows its statement.

Lemma 3. *Let A, s, r be positive integers, and D_s be a real number larger than 640 000. If we assume that every integer congruent to 3, 4, ..., 12 or 13 modulo 16 and congruent to r modulo 5 which lies in the interval $[A, A + D_s)$ is a B_s , then every integer congruent to $r + 1$ modulo 5 which lies in the interval $[16A, 16(A + D_{s+1}))$ is a B_{s+1} , where*

$$D_{s+1} = D_s + (D_s/40)^{4/3} \quad (\text{U-type ascent}).$$

6.3. End of the proof. Our starting point is the result (corollary) stated in §2, which asserts that all the integers in the interval $(m_1, m_2]$ which are congruent to 17 modulo 80 are B_7 's, with $m_1 = 36.5 \times 10^9$ and $m_2 = 9.59 \times 10^{18}$.

Our tools are the ascent steps S, T, and U. We have shown that the ascent sequences are among the following:

SSSSSSSS XXXX
 SSSSSSSXXXU
 SSSSSSSXXXUX

where an X step is of type S or T.

Since the T-type steps are less efficient than the S-type ones (cf. Lemma 2), we may replace X by T in order to compute the worst case. By appealing to Lemmas 2 and 3, a hand computation shows that any integer in the interval $(16m_1, 10^{448}]$ which is not congruent to 0 modulo 16 is a B_{19} .

It has been shown by hand that every integer in the range $(0, 16m_1]$ is a B_{19} (and this can be easily checked on a computer). Since we know that every integer which is not divisible by 16 in the interval $(0, 10^{448}]$ is a B_{19} , we finally conclude that every integer in the interval $(0, 10^{448}]$ is a B_{19} .

REMERCIEMENTS

Nous tenons à remercier vivement Henri Cohen et Michel Olivier pour l'aide très précieuse qu'ils nous ont apportée pour l'écriture et la mise au point des trois programmes.

Ce travail a bénéficié d'un soutien du PRC mathématique et informatique.

BIBLIOGRAPHY

1. R. Balasubramanian, J.-M. Deshouillers, and F. Dress, *Problème de Waring pour les bicarrés, 1: schéma de la solution*, C. R. Acad. Sci. Paris Sér. I **303** (1986), 85–88.
2. —, *Problème de Waring pour les bicarrés, 2: résultats auxiliaires pour le théorème asymptotique*, C. R. Acad. Sci. Paris Sér. I **303** (1986), 161–163.
3. J.-M. Deshouillers, *Le problème de Waring pour les bicarrés: le point en 1984*, Sémin. Theor. Analyt. Nb Paris, année 1984–85, exp. 33.
4. —, *Problème de Waring pour les bicarrés*, Sémin. Theor. Nb Bordeaux, année 1984–85, exp. 14.
5. —, *Sur la majoration des sommes de Weyl biquadratiques*, Ann. Scuola Norm. Sup. Pisa Cl. Sci. (4) **19** (1992), 291–304.
6. J.-M. Deshouillers and F. Dress, *Sommes de diviseurs et structure multiplicative des entiers*, Acta Arith. **49** (1988), 29–63.
7. —, *Sum of 19 biquadrates: on the representation of large integers*, Ann. Scuola Norm. Sup. Pisa Cl. Sci. (4) **19** (1992), 113–153.
8. G. H. Hardy and J. E. Littlewood, *Some problems of "Partitio Numerorum": VI. Further researches in Waring's problem*, Math. Z. **23** (1925), 1–37.
9. H. E. Thomas, *A numerical approach to Waring's problem for fourth powers*, Ph.D., The University of Michigan, 1973.
10. —, *Waring's problem for twenty two biquadrates*, Trans. Amer. Math. Soc. **193** (1974), 427–430.

ALGORITHMIQUE ARITHMETIQUE, UMR CNRS 9936, UNIVERSITÉ BORDEAUX 1, F-33405 TALENCE CEDEX, FRANCE

E-mail address: dress@mizar.greco-prog.fr

MATHÉMATIQUES STOCHASTIQUES, BP 26, UNIVERSITÉ BORDEAUX 2, F-33076 BORDEAUX CEDEX, FRANCE

E-mail address: dezou@frbdx11.bitnet