MATHEMATICS OF COMPUTATION
Volume 83, Number 288, July 2014, Pages 2033–2060
S 0025-5718(2013)02787-1
Article electronically published on November 18, 2013

EMPIRICAL VERIFICATION OF THE EVEN GOLDBACH CONJECTURE AND COMPUTATION OF PRIME GAPS UP TO $4 \cdot 10^{18}$

TOMÁS OLIVEIRA e SILVA, SIEGFRIED HERZOG, AND SILVIO PARDI

Abstract. This paper describes how the even Goldbach conjecture was confirmed to be true for all even numbers not larger than $4 \cdot 10^{18}$. Using a result of Ramaré and Saouter, it follows that the odd Goldbach conjecture is true up to $8.37 \cdot 10^{26}$. The empirical data collected during this extensive verification effort, namely, counts and first occurrences of so-called minimal Goldbach partitions with a given smallest prime and of gaps between consecutive primes with a given even gap, are used to test several conjectured formulas related to prime numbers. In particular, the counts of minimal Goldbach partitions and of prime gaps are in excellent accord with the predictions made using the prime k-tuple conjecture of Hardy and Littlewood (with an error that appears to be $O(\sqrt{t \log\log t})$, where t is the true value of the quantity being estimated). Prime gap moments also show excellent agreement with a generalization of a conjecture made in 1982 by Heath-Brown.

The Goldbach conjecture [13] is a famous mathematical problem whose proof, or disproof, has so far resisted the passage of time [20, Problem C1]. (According to [1], Waring and, possibly, Descartes also formulated similar conjectures.) It states, in its modern even form, that every even number larger than four is the sum of two odd prime numbers, i.e., that $n = p + q$. Here, and in what follows, n will always be an even integer larger than four, and p and q will always be odd prime numbers. The additive decomposition $n = p + q$ is called a Goldbach partition of n. The one with the smallest p will be called the minimal Goldbach partition of n; the corresponding p will be denoted by $p(n)$ and the corresponding q by $q(n)$.

It is known that up to a given number x at most $O(x^{0.879})$ even integers do not have a Goldbach partition [30], and that every large enough even number is the sum of a prime and the product of at most two primes [24]. Furthermore, according to [48], every odd number greater that one is the sum of at most five primes. As described in Table 1, over a time span of more than a century the even Goldbach conjecture was confirmed to be true up to ever-increasing upper limits. Section 1 describes the methods that were used by the first author, with computational help from the second and third authors, and others, to set the limit of verification of the Goldbach conjecture at $4 \cdot 10^{18}$. Section 2 presents a small subset of the empirical data that was gathered during the verification, namely, counts and first occurrences of primes in minimal Goldbach partitions, and counts and first occurrences of prime gaps, and compares it with the predictions made by

Received by the editor May 21, 2012 and, in revised form, December 6, 2012.
2010 *Mathematics Subject Classification.* Primary 11A41, 11P32, 11N35; Secondary 11N05, 11Y55.
Key words and phrases. Goldbach conjecture, prime gaps, prime k-tuple conjecture.

TABLE 1. Some records of verification of the even Goldbach conjecture.

limit	**year**	**who**
unknown	1742	Goldbach [13]
10^4	1855	Desboves [13] (confirmed by Haussner in 1896 [13])
10^5	1940	Pipping [44]
$3.3 \cdot 10^7$	1964	Shen [44]
10^8	1965	Stein and Stein [47] (confirmed by Light *et al.* in 1980 [28])
$2 \cdot 10^{10}$	1989	Granville, Van de Lune, and te Riele [19]
$4 \cdot 10^{11}$	1993	Sinisalo [46]
10^{14}	1998	Deshouillers, te Riele, and Saouter [11]
$4 \cdot 10^{14}$	2001	Richstein [40]
$3 \cdot 10^{17}$ (double checked)	2012	Oliveira e Silva, Herzog, and Pardi (this paper)
$4 \cdot 10^{18}$	2012	Oliveira e Silva, Herzog, and Pardi (this paper)

conjectured asymptotic formulas. It is also established there that the odd Goldbach conjecture, which states that every odd number larger than 5 is the sum of three primes, is true up to $8.37 \cdot 10^{26}$. Section 2.4 acknowledges those that contributed computational resources to this extensive verification effort.

1. Methods

To verify the even Goldbach conjecture for a given n two primes p and q must be found, possibly with q equal to p, such that $n = p + q$. Although any p for which $n - p$ is prime will do [11, 12, 44], we opted to compute for each n the minimal Goldbach partition $p(n) + q(n)$. The main reason for this choice is that the number of occurrences of a given smallest prime in a minimal Goldbach partition, as well as the smallest n for which it occurs, has some theoretical interest [19].

In order to compute the minimal Goldbach partitions for all even numbers belonging to a given interval it is necessary to have a list of the primes belonging to a possibly slightly larger interval; these primes will be the candidates for $q(n)$. Subsection 1.1 describes the modified segmented Eratosthenes sieve used to generate these primes. This modification, devised in 2001 when the computations reported in this paper were started, exhibits excellent data-cache behavior. Near 10^{18} our production code takes an average of about 10 clock cycles to determine if an odd number is prime or not.

Subsection 1.2 describes how the minimal Goldbach partition can be computed in a very efficient way for each even number belonging to a given interval. Irrespective of the order of magnitude of n, our production code takes an average of about 9 clock cycles to compute and collect statistics about each minimal Goldbach partition.

Subsection 1.3 describes how the computations were distributed among many computers. It also describes the measures that were taken in order to attempt to ensure that the computations were performed correctly. They were essential to locate occasional bad results due to random low probability hardware failures. Although very rare, such hardware failures are almost unavoidable in a computation that used a mixture of reliable and unreliable (low-cost personal computers) computing resources, and which took about 770 one-core CPU years to finish.

1.1. **Cache-efficient segmented Eratosthenes sieve.** Although several algorithms with better asymptotic computational complexity exist [2, 14, 17], the segmented Eratosthenes sieve [3, 5, 45] — with our own modifications — appears to be the fastest way to generate all primes in a relatively large interval with an upper limit near 10^{18}. This is so because the simplicity of the algorithm and its regular data requirements can be used to reduce the frequency of branch mispredictions and accesses to out-of-cache data, thus speeding up considerably the program on contemporary state-of-the-art general purpose processors. This is apparently not so easy to do with the other algorithms.

We begin with a description of the standard segmented Eratosthenes sieve and with an explanation of its shortcomings; p_k is the k-th prime number, i.e., $p_1 = 2$, $p_2 = 3$, and so on, $\lfloor x \rfloor$ is the largest integer not larger than x, $x \bmod y = x - y\lfloor \frac{x}{y} \rfloor$, and $\pi(x)$ denotes the number of primes not larger than x.

Algorithm 1.1 (Segmented Eratosthenes sieve [3])**.** *To generate all odd primes in the interval (A, B), with $B > A > 0$, with A even, with K and Δ integers, and with $B = A + 2K\Delta$, do:*

1. [Initialize.] *Set a to A and b to $A + 2\Delta$. Set j to 2.*
2. [New interval.] *Set $m_0, m_1, \ldots, m_{\Delta-1}$ to 1. Set i to 2.*
3. [New primes.] *If $p_j^2 \geq b$ then advance to step 5.*
4. *If $p_j^2 < a$ then set o_j to $\big(2p_j - 1 - (a + p_j) \bmod (2p_j)\big)/2$; otherwise set o_j to $(p_j^2 - a - 1)/2$. Add 1 to j and go back to step 3.*
   Comment: $a + 2o_j + 1$ is the smallest odd multiple of p_j larger than a that needs to be considered.
5. [Mark composites.] *If $i \geq j$ then advance to step 8.*
6. *If $o_i \geq \Delta$ then subtract Δ to o_i, add 1 to i, and go back to step 5.*
7. *Set m_{o_i} to 0. Add p_i to o_i. Go back to step 6.*
8. [Next interval.] *Add 2Δ to a and to b. If $a < B$ then go back to step 2; otherwise terminate.*

At the beginning of step 8, m_i is equal to 1 if and only if $a + 2i + 1$ is prime.

This algorithm requires that a list of the odd primes up to $\sqrt{B}$, plus the first prime larger than $\sqrt{B}$, to be available. Such a list can be computed easily with a simple modification of the same algorithm. It is possible to avoid storing the o_j variables; they can be recomputed every time a new (a, b) interval is being dealt with. Doing so, however, slows down the algorithm because divisions on contemporary processors are slow.

Under normal conditions only the inner (steps 6 and 7) and middle loops (steps 5 to 7) of Algorithm 1.1 are significant parts of the computation [3]. The number of times the middle loop is performed is

$$N_{\text{middle}} = \sum_{k=1}^{K} \pi\big(\sqrt{A + 2k\Delta}\big) - K \approx K\pi\big(\sqrt{B}\big)$$

(the approximation is valid when A is much larger that $B - A$, as is usually the case in practice). The number of times the inner loop is performed is, approximately

$$N_{\text{inner}} \approx \sum_{k=1}^{K} \sum_{2<p\leq\sqrt{a+2k\Delta}} \frac{\Delta}{p} \approx \frac{B-A}{2}\left(\log\log B - 0.93165\right)$$

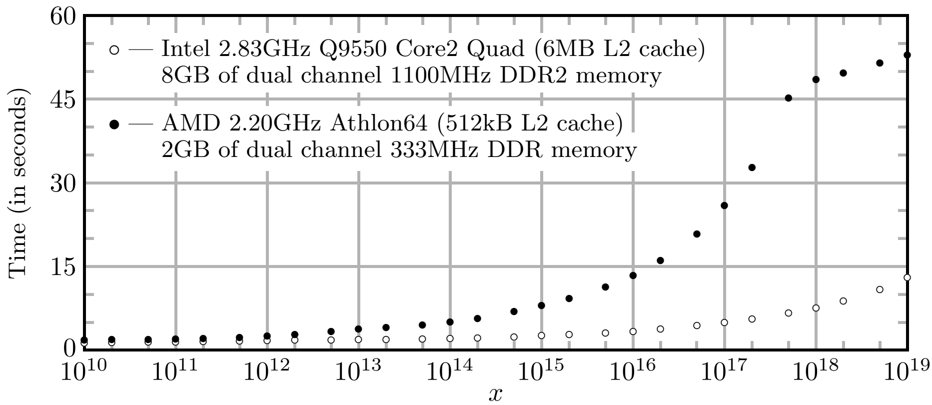

FIGURE 1. Time needed to generate all primes in an interval of 2^{30} integers centered at x using a simple implementation of Algorithm 1.1 [34, second program version], for two processors (only one core used on the Intel processor). The older single-core Athlon64 processor has a much smaller L2 cache, and slower main memory, which for large x makes the algorithm rather slow. For both processors, when x increases the optimal value of Δ also increases (not shown). The initialization time of the algorithm (steps 3 and 4 for the first interval), about a minute for the largest x on the slower processor, was not taken into consideration.

(the last approximation is a simple application of Mertens' second theorem [22]). The execution time of Algorithm 1.1 can then be reasonably well approximated by $\alpha_{\text{middle}}N_{\text{middle}}+\alpha_{\text{inner}}N_{\text{inner}}$, where α_{middle} and α_{inner} are constants that depend on the actual implementation of the algorithm and, of course, on the processor where it is run. The second term corresponds to the useful work made by the algorithm. The first corresponds to overheads and so should be made as small as possible. In the standard segmented Eratosthenes sieve this is achieved by making K small or, what is the same, by making Δ large [3]. Doing this, however, increases the amount of memory accessed in an essentially random way in the inner loop. If this amount of memory exceeds the amount that can be stored in the processor's data caches α_{inner} will be large and so the algorithm will be slow.

A small value of Δ, on the other hand, gives rise to a large value of K. In this case the algorithm spends a larger fraction of its time just updating the o_j variables. This is so because the middle loop is run more times and because the fraction of primes that have an odd multiple in the interval (a,b) decreases as b increases. For example, for $B = 10^{18}$ and $\Delta = 2^{19}$, only 0.553% (2 81049 in 508 47533) of the odd primes used to mark composites have an odd multiple belonging to the interval $(B-2\Delta,B)$. The best value for Δ will then be a trade-off between the need to make Δ small (to keep all frequently used variables in the data cache), and the need to make it large (to reduce the computational overheads). The end result is a program which slows down considerably when b increases beyond an implementation dependent limit, as illustrated in Figure 1.

There is a simple way to eliminate this problem. The main idea is to leave to later intervals all primes that do not have an odd multiple in the current interval.

In order to do this efficiently it is necessary to split the primes p_j in two classes: those that are smaller than Δ (the "small" primes), and those that are not (the "large" primes). The former are guaranteed to have at least one odd multiple in an interval of 2Δ consecutive integers, and can be dealt with as in Algorithm 1.1. The latter are guaranteed to have at most one odd multiple in such an interval (this observation was used in [3] to speedup the inner loop of Algorithm 1.1). To deal with them efficiently, the tuples (p_j, o_j) are placed in lists, one list per interval of the form $(A + k\Delta, A + (k+1)\Delta)$, in such a way that at the beginning of the middle loop of the algorithm the list associated with the current interval contains only the "large" primes which have an odd multiple in that interval. This idea gives rise to the following algorithm.

Algorithm 1.2 (Cache-efficient segmented Eratosthenes sieve)**.** *To generate all odd primes in the interval (A, B), with $B > A > 0$, with A even, with K and Δ integers, and with $B = A + 2K\Delta$, do:*

1. [Initialize.] *Set a to A and b to $A + 2\Delta$. Set k to 0, j to 2, and p to 3. Set the lists $L_0, L_1, \ldots,$ to the empty list.*
2. [New interval.] *Set $m_0, m_1, \ldots, m_{\Delta-1}$ to 1. Set i to 2.*
3. [New "small" primes.] *If $p \geq \Delta$ or if $p^2 \geq b$ then advance to step 5.*
4. *Set p_j to p. If $p^2 < a$ then set o_j to $\big(2p - 1 - (a+p) \bmod (2p)\big)/2$; otherwise set o_j to $(p^2 - a - 1)/2$. Add 1 to j and replace p by the smallest prime larger than p. Go back to step 3.*
5. [Mark composites.] *If $i \geq j$ then advance to step 8.*
6. *If $o_i \geq \Delta$ then subtract Δ to o_i, add 1 to i, and go back to step 5.*
7. *Set m_{o_i} to 0. Add p_i to o_i. Go back to step 6.*
8. [New "large" primes.] *If $p^2 \geq b$ then advance to step 10.*
9. *If $p^2 < a$ then set o to $\big(2p - 1 - (a + p) \bmod (2p)\big)/2$; otherwise set o to $(p^2 - a - 1)/2$. Insert the tuple $(p, o \bmod \Delta)$ in the list $L_{k+\lfloor o/\Delta \rfloor}$. Replace p by the smallest prime larger than p and go back to step 8.*
10. [Mark composites.] *For each tuple (p, o) of the list L_k, set m_o to 0 and insert the tuple $\big(p, (o + p) \bmod \Delta\big)$ in the list $L_{k+\lfloor (o+p)/\Delta \rfloor}$.*
11. [New interval.] *Set k to $k + 1$ and add 2Δ to a and to b. If $a < B$ then go back to step 2; otherwise terminate.*

At the beginning of step 11, m_i is equal to 1 if and only if $a + 2i + 1$ is prime.

On contemporary processors, the test at the beginning of step 6 generates many time-consuming branch mispredictions when p_j approaches Δ; in a practical implementation this can be ameliorated by dealing with the primes between, say, $\Delta/8$ and Δ (the "middle primes") in a way similar to how the "large" primes are handled. There is no such problem in step 10.

If there is enough space in the data caches to hold the m_i variables, the information where each list insertion point resides in memory, and one cache line for each active list, then the speed of the algorithm does not change much as b is increased, as illustrated in Figure 2.

An auxiliary sieve, updated using, for example, Algorithm 1.1, can be used to compute in an efficient way the sequence of the primes p used by Algorithm 1.2. The speed of both algorithms can be slightly improved by changing the way the variables m_i are initialized. For example, it is possible to set i to 7 in step 2 of both algorithms if the m_i variables are initialized with a precomputed pattern

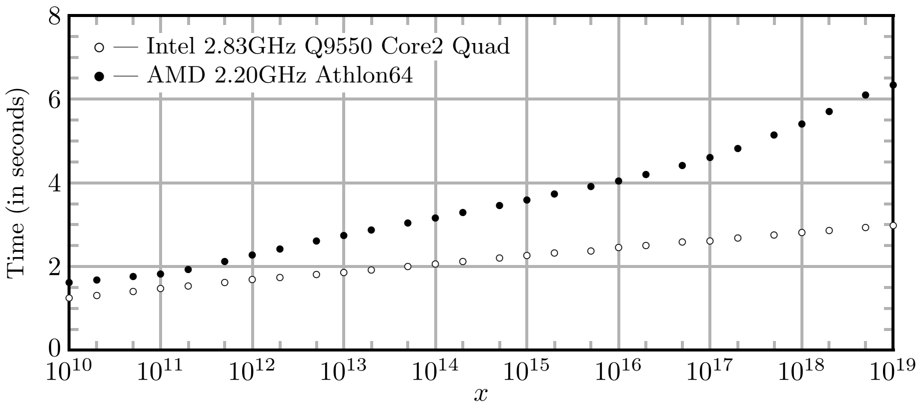

FIGURE 2. Time needed to generate all primes in an interval of 2^{30} integers centered at x using a simple implementation of Algorithm 1.2 [34, second program version] (see also [25]), for the two processors described in Figure 1 (only one core used on the Intel processor). The initialization time, about half a minute for the largest x on the slower processor, was not taken into consideration. For $x = 10^{19}$ this algorithm is about 8.4 times faster than Algorithm 1.1 on the Athlon64 and about 4.4 times faster on the Core2 Quad. Note that the improvement is larger on the processor with the smaller L2 cache.

determined by the first 5 odd primes (this pattern has a period of $3\times5\times7\times11\times13$). Of course, each m_i variable should be associated with a single memory bit.

In a practical implementation of Algorithm 1.2 the memory used by each list should grow as the need for it arises, i.e., it should be a linked list. Furthermore, at most $2+\lfloor\frac{\sqrt{B}}{\Delta}\rfloor$ linked lists can be non-empty at any given time. A circular buffer with a suitable size (a power of two is particularly useful) should then be used to store pointers to the insertion points of the linked lists. In order to use the data caches in an efficient way and to take advantage of the automatic memory prefetch mechanism of contemporary processors each linked list should be subdivided in relatively large chunks (each with, say, 4096 bytes of memory). The starting address of each chunk should be a multiple of the processor's data cache line size. Due to the large chunk size of each linked list component, the memory overhead needed to manage the linked lists is very small. Hence, the memory used by Algorithm 1.2 is only slightly larger than that used by Algorithm 1.1.

The single-threaded 32-bit prime generation code used in our empirical verification of the Goldbach conjecture is capable of generating primes up to $(30\times2^{26})^2\approx 4.05\cdot10^{18}$. It uses a modulo 30 wheel [37, 38] variant of Algorithm 1.2, i.e., only the numbers which are not multiples of 2, 3 and 5 are represented in the sieve. This complicates the algorithm but makes it almost twice as fast; near 10^{18} the average number of clock cycles required to determine if an odd integer is prime or not dropped from 14.8 to 8.7, and from 22.1 to 10.5, respectively, for the Core2 Quad and for the Athlon64 processors described in Figure 1. Assembly language was also extensively used.

TABLE 2. Empirical average value of i when Algorithm 1.3 terminates for intervals of the form $\big(10^{12}k, 10^{12}(k+1)\big)$.

k	i-average	$\frac{i\text{-average}}{\log(k+1/2)+12\log 10}$
1	15.58519	0.55589
10	16.67964	0.55631
100	17.93997	0.55643
1000	19.22367	0.55657
10000	20.51067	0.55673
1 00000	21.79939	0.55690
10 00000	23.08907	0.55708

1.2. **Computation of the minimal Goldbach partition of all even numbers belonging to a given interval.** We begin by presenting a simple algorithm, capable of computing the minimal Goldbach partition of a single even number n. It will be used by a more efficient algorithm, presented below, to deal with the (rare) cases not dealt with by that algorithm.

Algorithm 1.3 (Computation of the minimal Goldbach partition of n)**.** *To compute the minimal Goldbach partition $n = p(n) + q(n)$, do:*

1. [Initialize.] *Set i to 2.*
2. [Test.] *If $2p_i > n$ then terminate, stating that there is no Goldbach partition of n.*
3. *If $n - p_i$ is prime, then set $p(n)$ to p_i and $q(n)$ to $n - p_i$, and terminate.*
4. [Try next prime.] *Increase i and go back to step 2.*

It was found empirically that the average value of i when this algorithm terminates (successfully) is approximately $0.557 \log n$ (cf. Table 2). This, and the clock cycles lost due to a branch misprediction that is usually present when the algorithm terminates makes it too slow to be used in the computation of the minimal Goldbach partition of all even integers belonging to a large interval. That can be done efficiently using a segmented version (not presented) of the following algorithm.[1]

Algorithm 1.4 (Computation of the minimal Goldbach partition of all even numbers belonging to an interval)**.** *To compute the minimal Goldbach partition for all even numbers belonging to the interval (C, D), with C and D odd, do:*

1. [Initialize.] *Set I to a value that depends on D and on the processor model (see below). Set J to $(p_I + 1)/2$. Set L to $(D - C)/2$. Set u_0, u_1, …, u_{L+J-1} to zero.*
   Comment: u_i will contain information about the smallest prime in the minimal Goldbach partition of $C + 1 + 2i$.
2. [Mark.] *For each prime q belonging to the interval $(C - 3, D - 3)$, ordered in increasing order, do step 3 (a subroutine) with j set to $(q - C)/2$. After all primes q have been dealt with, go to step 4.*
3. *For $i = 2, 3, \ldots, I$, set k to $j + (p_i - 1)/2$ and then set u_k to i.*
   Comment: u_k may be updated latter with a smaller i value (larger q prime).

[1] We rediscovered this way of speeding up Algorithm 1.3. Haussner used a similar idea to speed up the construction of Goldbach partition tables up to 10^4 [1]. The algorithms used in [19, 40, 46] only compute the minimal Goldbach partition when $p(n)$ is larger than an implementation-defined limit; also, they loop on n and not on q.

TABLE 3. Best average number of clock cycles (T_{avg}) used by Algorithm 1.4 to compute $p(n)$, and to collect statistical data, for an even integer near x, and the corresponding best value of the I parameter for two different processor models (cf. Figure 1); for the Core2 Quad $I \approx 2.50 \log x - 13.7$, and for the Athlon64 $I \approx 2.83 \log x - 12.4$.

	Core2 Quad			**Athlon64**		
x	T_{avg}	I	$\frac{I+13.7}{\log x}$	T_{avg}	I	$\frac{I+12.4}{\log x}$
10^{12}	9.837	56	2.523	8.234	66	2.837
10^{13}	9.788	61	2.496	8.238	72	2.820
10^{14}	9.746	67	2.503	8.212	79	2.835
10^{15}	9.714	72	2.481	8.195	85	2.820
10^{16}	9.707	78	2.489	8.210	92	2.834
10^{17}	9.701	84	2.496	8.207	98	2.820
10^{18}	9.707	90	2.502	8.226	105	2.833

4. [Finish.] *For* $i = 0, 1, \ldots, L-1$*, set* n *to* $C + 1 + 2i$*; if* u_i *is not zero then set* $p(n)$ *to* p_{u_i}*; otherwise compute* $p(n)$ *using Algorithm* 1.3 *(with* i *set to* $I + 1$ *in its first step). Set* $q(n)$ *to* $n - p(n)$*.*

In other words, for each prime q belonging to the interval $(C-3, D-3)$ one updates the array u in the positions corresponding to the even integers $3+q$, $5+q$, …, $p_I + q$ with the values 2, 3, …, I. In the end, the number stored in each array position will be either zero, if no Goldbach partition was generated for the even number corresponding to that position, or the index of the smaller prime of the last Goldbach partition that was generated for that even integer (it will be the minimal Goldbach partition if the primes q are processed in increasing order). In the former case the minimal Goldbach partition has to be computed using Algorithm 1.3.

It turns out that the choice $I = \lfloor \alpha \log D + \beta \rfloor$, with α and β parameters that depend on the processor model, approximately minimizes the execution time of the algorithm. This is illustrated in Table 3, which presents best I values and the corresponding average number of clock cycles per even integer used by our most efficient implementation (in assembly) of a segmented version of Algorithm 1.4 for the two processors described in Figure 1. Remarkably, the average number of clock cycles remains practically constant. This is so because for the best I the amount or work done in steps 2 and 3 of Algorithm 1.4 is approximately given by $(D-C)(\alpha + \beta/\log D)$, i.e., it does not change much with D when $D - C$ is held constant, and because for the best I the relative frequency that Algorithm 1.3 is invoked in step 4 of Algorithm 1.4 is approximately inversely proportional to $\log D$.

In order to make Algorithm 1.4 as fast as possible, the loop of step 3 should be unrolled. In our final implementation when the computation starts, self-modifying assembly code is used to trim this unrolled loop to the appropriate value of I. Furthermore, each loop iteration is performed by a single move immediate instruction, using the base register plus constant offset addressing mode (depending on the processor, up to two such instructions can usually be executed in each clock cycle). If I is large enough, then in step 4 u_i will be non-zero with a relative frequency close to one. The test "u_i is not zero" will then not be mispredicted often by the processor, and the slower Algorithm 1.3 will be invoked rarely.

1.3. **Computational details and error detection and correction measures.** Our code was developed in 2001 for Intel/AMD (x86 instruction set) single-core 32-bit processors. Although later a 64-bit instruction set for AMD/Intel processors appeared, given the initial large investment in both the optimization (assembly language, software pipelining) and in the verification of the correctness of the code (the output of each assembly language routine was compared to the output of a slower `C` language routine that used a simpler fool-proof algorithm), it was deemed prudent to not produce a 64-bit version of the code. Given the programming techniques used, it was estimated that a 64-bit version would be a few percent faster that a 32-bit version.

The entire computation was split into disjoint intervals of 10^{12} integers; the k-th interval, $0 \le k < 4 \cdot 10^6$, covers the even integers that satisfy the conditions $\max(4, 10^{12}k) < n \le 10^{12}(k+1)$. Testing each interval required between eight hours (in the year 2001) and about forty minutes (in the year 2012). Processors with more than one core can test in parallel, with a very mild degradation in performance, a number of intervals equal to the number of cores they have. On Intel processors with hyper-threading capabilities, testing two intervals on the same processor core takes between 50% (core i7) and 80% (core i3) more time than testing a single interval on that core (a gain between 2/1.5 and 2/1.8).

A master-worker paradigm was used to automatically manage the computations: a central master, used to distribute the intervals among a pool of workers and to collect the data of processed intervals, and many workers that did the actual testing work. Each worker had a unique ID and was capable of processing several intervals without contacting the master. Intervals not processed within a prespecified time limit were redistributed to other workers. Windows and GNU/Linux versions of the worker code were produced (to ensure correctness, the low-level functions were exactly the same in the two cases). A worker was also capable of working without a master; that capability was used on high-performance computing environments. In those cases, the distribution of the intervals and collection of results was done using semi-automatic tools specially developed for that purpose.

The data computed and recorded for each interval of 10^{12} integers includes:

- two worker IDs (intervals can be double checked by workers with different IDs), and the respective number of seconds that were used to process them,
- counts of the number of primes in each of the 32 primitive residue classes modulo 120,
- counts and the first occurrence of minimal Goldbach partitions with a given smallest prime,
- counts and the first occurrence of gaps between prime numbers, and
- a 32-bit cyclic redundancy check sum.

(Due to an unfortunate oversight, a high-precision approximation to the sum of the inverses of the twin primes was not collected.) The entire data was stored in 4000 files, each holding information about 1000 intervals, using a total of about 27GB of storage space.

The processed data of an interval received from a worker was screened by the master to detect obvious errors: the sum of the counts of minimal Goldbach partitions had to match the number of even numbers belonging to the interval, and the sum of the counts of prime gaps had to match the sum of the primes in the residue classes modulo 120. These two tests never failed. The following offline screening test was then performed for each interval of 10^{12} integers: the computed number of primes belonging to the interval was compared to an independent count obtained using the first author's implementation of a combinatorial method to compute $\pi(x)$ [8, 27, 35] (this extra data was generated using about 20 one-core CPU years). It turned out that this test was very good at detecting bad results. This happened on a few occasions in the early stages of the computation (and very, very rarely later on), when personal computers, in particular, their memory subsystems, were less reliable than those that can be bought in 2012 (when the computations reported in this paper were finished). Once a bad result was detected the entire interval was recomputed, the computer that produced it was black-listed, and all intervals previously processed by that computer were double-checked. This procedure did not uncover more bad results.

Some time after the verification limit of 10^{18} was reached, the number of primes in the residue classes modulo 4 reported in [9] was compared to those counted in our verification efforts. To our dismay, a discrepancy of one was found in two of the residue classes between $3 \cdot 10^{17}$ and $4 \cdot 10^{17}$. Fortunately, Mark Deléglise's program was publicly available. Using it, a bisection strategy allowed us to locate quickly the interval with the bad result. This was dealt with as described at the end of the previous paragraph. To reduce considerably the probability of a (very rare) error of this kind to remain undetected, a final screening test was performed, this time for each interval of 10^{15} integers: the counts of the primes in the residue classes modulo 120 were compared to the counts obtained using Deléglise's program (this extra data was generated using about 10 one-core CPU years). No further discrepancies were detected.

As a final precaution, the entire interval up to $3 \cdot 10^{17}$ was double-checked, and the intervals containing one of the first 100 occurrences of a smallest prime in a minimal Goldbach partition or of a prime gap, as well as about 4% of the remaining intervals were also double-checked. No further discrepancies were detected. As expected, no errors were ever found on computations done on high-performance computing environments (they account for about 25% of all our data). We are therefore highly confident that all of our counts and first occurrences are correct. We feel that further double-checks are best left for a future still larger verification effort.

2. Results

In this section we present some results extracted from the data collected by our confirmation of the truth of the even Goldbach conjecture up to $4 \cdot 10^{18}$. In subsection 2.1 we present record values of first and late first occurrences of a prime in a minimal Goldbach partition, test the conjecture [19] that $p(n) = O(\log^2 n \log\log n)$, and compare the number of occurrences of a given prime in the minimal Goldbach partitions up to $4 \cdot 10^{18}$ with predictions made using the inclusion-exclusion principle applied to the prime k-tuples conjecture [21]. In subsection 2.2 we do the same, but for prime gaps (testing this time the conjecture [7, 18, 43] that $p_{n+1} - p_n = O(\log^2 n)$). In subsection 2.3 we compare prime gap moment data with corresponding predictions made by a conjecture of Heath-Brown [23]. Finally, in subsection 2.4 it is shown that our new verification limit of the even Goldbach conjecture can be used to prove without extra computation that the odd Goldbach conjecture is true up to $8.37 \cdot 10^{26}$.

TABLE 4. Record-breaking values of $p(n)$ for $n \leq 4 \cdot 10^{18}$.

n	$p(n)$	n	$p(n)$	n	$p(n)$
6	3	107 59922	829	834 29455 44436	3917
12	5	241 06882	929	1059 16059 00482	4003
30	7	277 89878	997	1298 22701 97518	4027
98	19	379 98938	1039	1519 79009 94218	4057
220	23	601 19912	1093	2899 80506 50046	4327
308	31	1136 32822	1163	4687 84427 66282	4519
556	47	1878 52862	1321	7690 35744 97118	4909
992	73	3350 70838	1427	18416 24778 60248	5077
2642	103	4199 11924	1583	21736 13167 06568	5209
5372	139	7210 13438	1789	38996 50268 19938	5569
7426	173	18471 33842	1861	1 04761 05758 36828	6469
43532	211	74732 02036	1877	6 25326 23459 30828	6961
54244	233	1 10010 80372	1879	24 92555 60081 75266	7559
63274	293	1 27039 43222	2029	31 28417 79105 28922	7753
1 13672	313	2 12485 58888	2089	121 00502 23040 07026	8443
1 28168	331	3 58840 80836	2803	255 32912 66885 55994	8501
1 94428	359	10 59638 12462	3061	258 54942 69161 49682	8933
1 94470	383	24 48855 95672	3163	555 27435 15567 50822	8941
4 13572	389	59 95335 46358	3457	887 12380 30778 37868	9161
5 03222	523	313 20592 94006	3463	906 03057 95622 79642	9341
10 77422	601	362 08211 73302	3529	2795 93511 65744 69638	9629
35 26958	727	443 83276 72994	3613	3325 58170 73339 60528	9781
38 07404	751	532 05038 15888	3769		

2.1. **Minimal Goldbach partitions.** As in [19], let $S(p)$ be the smallest even integer n for which $p(n) = p$ and let $L(p, x)$ be the number of even integers not larger than x for which $p(n) = p$. Table 4 presents the record-breaking values of $p(n)$, i.e., values of $p(n)$ larger than those for all smaller values of n (sometimes also called maximal values), that were found in this verification. It extends Table 3 of [4], Table 3 of [19], Table 1 of [46], and Table 1 of [40]. Table 5 presents the record-breaking values of $S(p)$ that were found. It extends Table 2 of [46].

2.1.1. *Conjectures concerning $p(n)$ bounds.* In [19] it was conjectured that $p(n) = O(\log^2 n \log\log n)$. In an email exchange in April 2012, Andrew Granville, using probabilistic arguments, suggested to the first author two more precise (incompatible) conjectures, both of the form $p(n) \leq \big(C + o(1)\big) \log^2 n \log\log n$: one with $C = C_2^{-1} \approx 1.51478$ and another, using a more refined argument, with $C = 2e^{-\gamma} C_2^{-1} \approx 1.70098$, where $C_2 \approx 0.66016$ is the twin primes constant and where $\gamma \approx 0.57722$ is Euler's constant. To test these conjectures, Figure 3 presents a plot of the values of

$$Q_1(p) = \frac{p}{\log^2 S(p) \log\log S(p)}$$

that we were able to compute. For our data $Q_1(p)$ clearly stays below 1.7 and only two points lie above 1.514: $Q_1(3) \approx 1.60231$ and $Q_1(6469) \approx 1.52627$. As explained in subsubsection 2.1.3, our empirical $L(p, x)$ data suggests that the slowly increasing trend that can be observed in Figure 3 will not persist for ever. Given that these conjectures allow a finite number of solutions of $Q_1(p) > C + \epsilon$, and taking into

TABLE 5. Record-breaking values of $S(p)$ for $S(p) \le 4 \cdot 10^{18}$.

p	$S(p)$	p	$S(p)$	p	$S(p)$
3	6	1049	3794 10652	4133	21528 58899 79816
5	12	1061	5544 63808	4241	28078 06211 53342
7	30	1091	6785 46502	4373	31924 55155 37554
11	124	1097	11688 88534	4457	33263 84502 61204
17	418	1283	16732 68292	4523	44568 54135 00946
37	1274	1301	19275 28888	4621	55724 95547 49362
53	2512	1327	23314 65314	4643	65020 45060 20934
59	3526	1429	25388 33642	4679	66695 60251 01272
71	4618	1439	28165 93312	4721	81441 08625 37738
83	7432	1451	44071 65118	4733	1 02520 38425 12482
89	12778	1493	58018 28806	4817	1 24657 87228 03144
101	26098	1559	89466 30856	4937	1 84205 42851 36636
131	34192	1571	2 14399 65412	5051	2 30360 82907 75108
149	37768	1787	2 60702 02114	5087	2 74844 32963 52086
167	59914	1811	3 03257 42068	5227	3 77167 15201 32578
179	88786	1867	3 08343 71756	5333	4 46303 92199 37862
191	97768	1873	3 26526 27542	5471	5 12249 86761 96358
197	1 12558	1889	4 44603 16708	5483	6 19847 81686 28056
223	2 21942	1907	6 42439 62808	5501	14 21174 44030 75144
257	2 37544	1997	6 53347 25368	5879	15 81237 99596 45512
263	4 85326	2027	11 38431 30358	5903	20 01798 63813 70774
281	6 42358	2153	24 48089 93116	5987	31 82162 58292 50454
317	6 86638	2351	38 46192 17512	6131	48 03378 79780 24768
347	10 42078	2441	74 38910 46202	6263	55 10400 89583 65746
379	11 72918	2459	83 88139 74892	6491	107 15720 71018 94788
401	20 41402	2663	157 80847 23724	6761	182 74530 72010 20658
419	24 06448	2837	254 12467 52056	6899	237 09861 61937 22886
463	42 88574	2963	322 83172 20754	7013	296 54004 27271 13116
487	49 38848	2969	604 65005 99278	7187	344 20574 38160 95468
509	92 92156	3023	711 95508 17194	7307	370 58110 67669 09188
521	143 41888	3137	740 55675 22324	7489	411 41162 99917 22966
569	177 26098	3203	1077 03538 52014	7577	558 61954 75699 07716
593	207 57292	3323	1745 51588 97256	7649	754 27622 88329 57188
659	325 07242	3449	1856 69525 90488	7691	813 69562 21921 68004
739	343 62758	3557	3636 14483 59204	7703	1473 61172 23318 22212
743	378 90844	3659	3902 83776 47218	7853	1599 56602 59143 18344
761	493 58128	3677	4085 46803 72224	7949	1793 16778 59048 03016
773	687 88066	3701	4477 67061 82504	8039	2043 43718 01888 10768
839	1297 96642	3761	5413 30158 34948	8087	2758 16342 81002 38178
853	1445 16902	3863	6091 30487 45092	8243	3244 40008 45058 12356
911	1503 86932	3923	10325 23255 78522	8273	3511 79756 73597 60604
941	2068 92484	4073	12998 77000 25542	8369	3714 75979 38306 49402
977	2470 13164	4079	14352 12522 89068	8387	3878 29701 74376 46306
1031	2994 34108	4127	19453 91791 43308	8423	$> 4 \cdot 10^{18}$

consideration the logarithmic scale associated to this problem, it seems likely that much more data (up to 10^{100} or even more) will be needed to empirically determine C directly with some accuracy, and hence determine which of the two conjectures is more plausible.

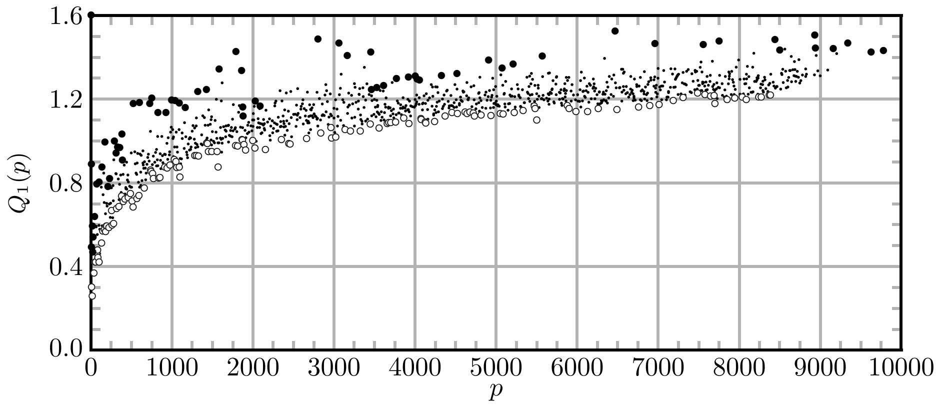

FIGURE 3. Plot of $Q_1(p)$ for $S(p) \leq 4 \cdot 10^{18}$. Disks ($\bullet$), circles ($\circ$), and dots ($\cdot$) correspond respectively to data obtained from Table 4, from Table 5, and to values of $S(p)$ that did not make it to either of the two tables.

2.1.2. *Estimate of $L(p,x)$ using the prime k-tuple conjecture.* Let $\boldsymbol{h} = \{h_1, \ldots, h_k\}$ be a set of k distinct integers, all of the same parity, and let $\pi(x;\boldsymbol{h})$ be the number of k-tuples $(m+h_1,\ldots,m+h_k)$, with $1 \leq m \leq x$, containing only primes. By the inclusion-exclusion principle

$$L(p,x) = -\sum_{\boldsymbol{s}} (-1)^{|\boldsymbol{s}|} \pi(x;\boldsymbol{s}), \tag{2.1}$$

where the sum is over all subsets $\boldsymbol{s}$ of $\{\,-3,-5,-7,-11,\ldots,-p\,\}$ which contain $-p$, and where $|\boldsymbol{s}|$ denotes the cardinality of $\boldsymbol{s}$. In [21] Hardy and Littlewood conjectured, with $c = 2$, that

$$\pi(x;\boldsymbol{h}) \sim G(\boldsymbol{h}) \int_c^x \frac{dt}{\log^k t}, \tag{2.2}$$

where

$$G(\boldsymbol{h}) = 2^{k-1} \prod_p \left(1 - \frac{\nu_p(\boldsymbol{h})}{p}\right) \left(1 - \frac{1}{p}\right)^{-k} \tag{2.3}$$

and where $\nu_p(\boldsymbol{h})$ is the number of distinct residue classes modulo p occupied by the elements of $\boldsymbol{h}$. Using this so-called prime k-tuple conjecture to approximate $\pi(x;\boldsymbol{s})$ in (2.1) yields

$$\hat{L}(p,x) = \sum_{k=1}^{\pi(p)-1} (-1)^{k+1} C_{p,k} \int_c^x \frac{dt}{\log^k t}, \tag{2.4}$$

where $C_{p,k} = \sum_{|\boldsymbol{s}|=k} G(\boldsymbol{s})$. The $C_{p,k}$ constants can be computed using a simple adaptation of the method used in [6] to compute other constants of the same kind. The first author computed them all for $p < 250$ using about 16 one-core CPU months. As an example of the general behavior of these constants, Table 6 presents the non-zero values of $C_{241,k}$.

It turns out that for relatively small values of x the lower limit of integration of 2 suggested by Hardy and Littlewood for (2.2) is a very bad choice for (2.4) when

TABLE 6. Non-zero values of $C_{241,k}$ (only 21 significant digits shown).

k	$C_{241,k}$	k	$C_{241,k}$
1	1.00000 00000 00000 00000	23	$3.70204\,14661\,49439\,69979 \cdot 10^{24}$
2	$1.13158\,66859\,65499\,59139 \cdot 10^{2}$	24	$1.31461\,87368\,38578\,84258 \cdot 10^{25}$
3	$6.93019\,94386\,60869\,24137 \cdot 10^{3}$	25	$4.22256\,92828\,55965\,63028 \cdot 10^{25}$
4	$3.00886\,99646\,95696\,40719 \cdot 10^{5}$	26	$1.22482\,06601\,55783\,90143 \cdot 10^{26}$
5	$1.01640\,78939\,12162\,69790 \cdot 10^{7}$	27	$3.20204\,63609\,01368\,30154 \cdot 10^{26}$
6	$2.79258\,80742\,87881\,31431 \cdot 10^{8}$	28	$7.52660\,46955\,07176\,17022 \cdot 10^{26}$
7	$6.41741\,99060\,14428\,77794 \cdot 10^{9}$	29	$1.58609\,30493\,60132\,31281 \cdot 10^{27}$
8	$1.25913\,74972\,52254\,51935 \cdot 10^{11}$	30	$2.98608\,21872\,48675\,88621 \cdot 10^{27}$
9	$2.14288\,09248\,75761\,71467 \cdot 10^{12}$	31	$5.00143\,91728\,42627\,39468 \cdot 10^{27}$
10	$3.20144\,28071\,44559\,73700 \cdot 10^{13}$	32	$7.41494\,33404\,69631\,24282 \cdot 10^{27}$
11	$4.23668\,72148\,78062\,42359 \cdot 10^{14}$	33	$9.67103\,74237\,26498\,34947 \cdot 10^{27}$
12	$4.99990\,15938\,62122\,23271 \cdot 10^{15}$	34	$1.10137\,98332\,87079\,45198 \cdot 10^{28}$
13	$5.28865\,62801\,63349\,25545 \cdot 10^{16}$	35	$1.08516\,76207\,67543\,49852 \cdot 10^{28}$
14	$5.03316\,43841\,95479\,05620 \cdot 10^{17}$	36	$9.14471\,69789\,56584\,84128 \cdot 10^{27}$
15	$4.32228\,77040\,16020\,86166 \cdot 10^{18}$	37	$6.49627\,26305\,32786\,34274 \cdot 10^{27}$
16	$3.35672\,30146\,84477\,12695 \cdot 10^{19}$	38	$3.81830\,48373\,21482\,47613 \cdot 10^{27}$
17	$2.36124\,94061\,35894\,65715 \cdot 10^{20}$	39	$1.81159\,68041\,87622\,69166 \cdot 10^{27}$
18	$1.50615\,10047\,65390\,73306 \cdot 10^{21}$	40	$6.70676\,47470\,80130\,86245 \cdot 10^{26}$
19	$8.71726\,56912\,30150\,63187 \cdot 10^{21}$	41	$1.84470\,56245\,09659\,86010 \cdot 10^{26}$
20	$4.57924\,55341\,30673\,89384 \cdot 10^{22}$	42	$3.49098\,59394\,38777\,29499 \cdot 10^{25}$
21	$2.18311\,41710\,01000\,00195 \cdot 10^{23}$	43	$3.96213\,08971\,56314\,45799 \cdot 10^{24}$
22	$9.44187\,69191\,50547\,38724 \cdot 10^{23}$	44	$1.95366\,73527\,22360\,22383 \cdot 10^{23}$

accurate estimates are desired. For example, using $c = 2$ we get $\hat{L}(241, 10^4) \approx -4 \cdot 10^{24}$, which is very far from its true value of zero, while using $c = 0$ we get $\hat{L}(241, 10^4) \approx -1.23592$, which is a much more reasonable estimate. Using $c = p$ we get $\hat{L}(241, 10^4) \approx 0.00084$, which is again a very reasonable estimate.[2] The same behavior was observed of all other values of p and of x that were tried. Therefore, for simplicity of computation, in all of our comparisons between $L(p, x)$ and $\hat{L}(p, x)$ a lower limit of integration of $c = 0$ was used. Furthermore, as illustrated in Table 7 for $x = 4 \cdot 10^{18}$ and $p = 241$, most of the non-zero $C_{p,k}$ constants are important (for x large enough all will be important).

Inspired by formula 5 of [7], which results from the application of the law of the iterated logarithm [15] to a random counting function that attempts to mimic the large scale behavior of $\pi(x)$, it was decided to test the possibility that the large deviation behavior of $\hat{L}(p, x) - L(p, x)$ follows a similar law. Considering that it is reasonable to expect that prime number patterns follow, asymptotically, a Poisson distribution [16, 26], which implies that variances should be equal to means, one may expect that $\left|\hat{L}(p, x) - L(p, x)\right|$ exceeds $(1 + \epsilon)\sqrt{2L(p, x) \log\log L(p, x)}$ at most a finite number of times. However, the law of the iterated logarithm assumes that

[2] It is necessary to avoid a lower limit of integration near 1, because $\hat{L}(p, x)$ blows up in that case (the principal values of the integrals present in (2.4) are used when $c < 1$ and $x > 1$). It is remarkable that, for $c = 0$, $\left|\hat{L}(p, p)\right| < 6$ for $p < 250$. (We have no explanation for this behavior; it implies an almost perfect cancellation of the large terms in the finite alternating series (2.4).) Thus, both $c = 0$ and $c = p$ are reasonable lower integration limits ($c = 2$ in not), at least for $p < 250$. The partial sums of (2.4) appear to converge faster when $c = 0$ than when $c = p$. The choice $c = 0$ has the added advantages of being more natural and being constant.

TABLE 7. Approximation of $L(p,x)$ by truncation of $\hat{L}(p,x)$ to K terms, for $c = 0$, $p = 241$, and $x = 4 \cdot 10^{18}$.

K	$\hat{L}(p,x)$	K	$\hat{L}(p,x)$
1	95 67626 09731 64698.5	10	6 29668 65710 23021.9
2	−163 45040 70427 19193.7	15	8 30450 15601 29840.7
3	216 63550 93958 03246.5	20	8 30304 11009 71376.4
4	−178 87696 79611 98263.7	25	8 30304 11896 68030.5
5	141 58277 14924 86186.5	30	8 30304 11896 67526.0
6	−69 77094 80643 09200.0	44	8 30304 11896 67526.0
		$L(p,x)$	8 30304 11498 24931

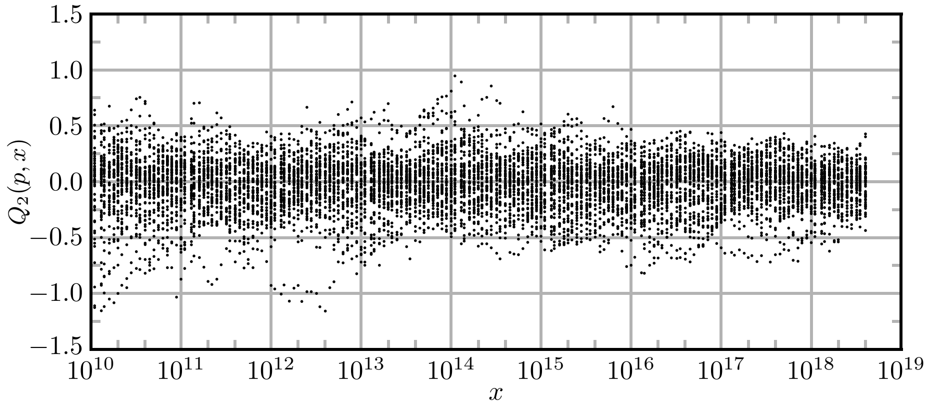

FIGURE 4. Plot of $Q_2(p,x)$ for $p < 250$ and for some values of x.

the random variables are independent, which is not the case here, so the above bound may not be correct. Nonetheless, one may hope that it captures the correct order of magnitude of the error term. To test this, Figure 4 presents a plot of some values of

$$Q_2(p,x) = \frac{\hat{L}(p,x) - L(p,x)}{\sqrt{2L(p,x)\log\log L(p,x)}},$$

for $p < 250$ and for selected values of x between 10^{10} and $4\cdot 10^{18}$ (twenty per decade, approximately equispaced on a logarithmic scale). From this figure it appears that $\bigl|Q_2(p,x)\bigr|$ may indeed be bounded (if not its growth rate should be very, very small). It also appears that the factor of two inside the square root may be slightly too large. These empirical observations suggest that, asymptotically, one should have

$$\bigl|\hat{L}(p,x) - L(p,x)\bigr| = O\left(\sqrt{\frac{x\log\log x}{\log x}}\right)$$

(since $C_{p,1} = 1$ one has $\hat{L}(p,x) \sim \frac{x}{\log x}$, and so one should also have $L(p,x) \sim \frac{x}{\log x}$).

2.1.3. *Rate of decay of* $L(p,x)$. It appears that, on a logarithmic scale, $L(p,x)$ does not deviate much from $\pi(x)\exp\bigl(-(\pi(p)-2)/(0.755\log x - 4.19)\bigr)$. This empirical result was obtained by first using best least-squares fits to approximate $\log L(p,x)$ by $m_1(x)\pi(p) + b_1(x)$ for several values of x between 10^{10} and $4 \cdot 10^{18}$ (discarding

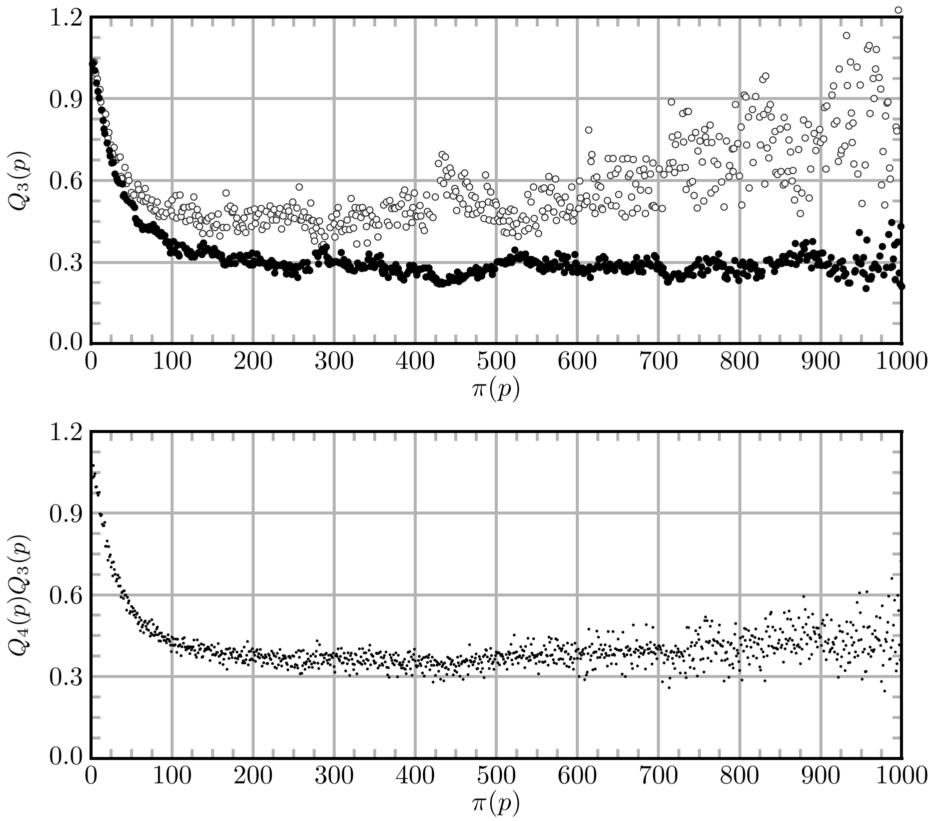

FIGURE 5. Plot of $Q_3(p)$ and of $Q_4(p)Q_3(p)$, for $2 \le \pi(p) \le 1000$, i.e., for $3 \le p \le 7919$. On the plot of $Q_3(p)$ the points with $p \bmod 3 = 1$ are represented by circles (∘) and the rest by disks (•).

data points as soon as $L(p,x) < 100$), and then by using another best least-squares fit to approximate $1/m_1(x)$ by $m_2 \log x + b_2$ (this last fit was extremely good). To study the deviations of the decay of $L(p,x)$ from a true exponential decay, the upper part of Figure 5 presents a plot of some values of

$$Q_3(p) = 10^{-17} e^{0.0355\pi(p)} L(p, 4 \cdot 10^{18}).$$

The factor $e^{0.0355\pi(p)}$ removes most of the exponential decay of $L(p, 4 \cdot 10^{18})$. The scale factor $10^{-17} \approx 1/\pi(4 \cdot 10^{18})$ places $Q_3(p)$ close to 1. Similar behavior was observed for other values of x (with different exponents and scale factors). The ups and downs of the $p \bmod 3 = 1$ points (∘) and of the $p \bmod 3 = 2$ points (•) are closely connected to what is happening to the difference $\Delta(p) = \pi(p;3,2) - \pi(p;3,1)$, where $\pi(x;m,a)$ denotes the number of primes up to x congruent to a modulo m. The extra factor

$$Q_4(p) = \begin{cases} 1 - 0.04\Delta(p), & \text{if } p \bmod 3 = 1, \\ 1 + 0.04\Delta(p), & \text{if } p \bmod 3 \ne 1, \end{cases}$$

approximately removes most of the fluctuations of $Q_3(p)$, as can be observed in the lower part of Figure 5 (the constant 0.04 was found by trial and error). Section 5

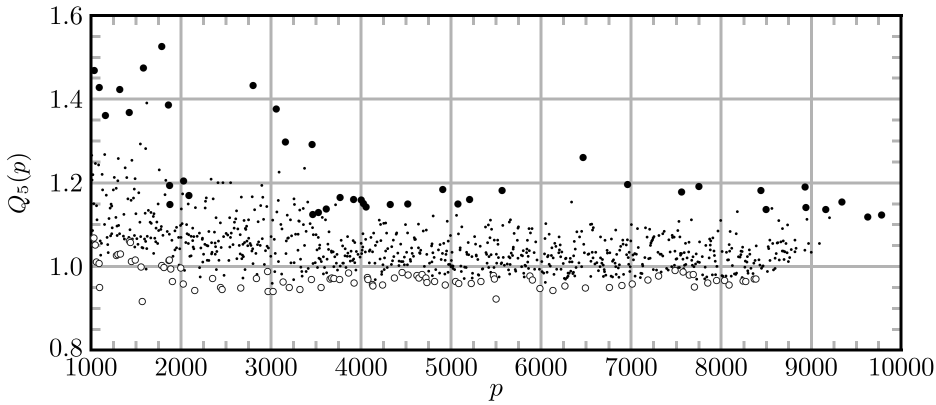

FIGURE 6. Plot of $Q_5(p)$ for $S(p) \leq 4 \cdot 10^{18}$ and for $p > 1000$. Disks (•), circles (○), and dots (·) correspond respectively to data obtained from Table 4, from Table 5, and to values of $S(p)$ that did not make it to either of the two tables.

of [19] provides an heuristic explanation for this last empirical observation. We were unable to explain the residual pattern observed in the lower part of Figure 5.

It is reasonable to expect that the first occurrence of a minimal Goldbach partition with $p(n) = p$ has an order of magnitude similar to that of the solution of $\hat{L}(p, x) = 1$ (this is indeed the case for $p < 250$). From our observed approximate exponential decay of $L(p, x)$ it then follows that it is likely that $S(p)$ has an order of magnitude similar to that of the solution of

$$\pi(x) \exp\left(-\frac{\pi(p) - 2}{0.755 \log x - 4.19}\right) = 1. \tag{2.5}$$

The left-hand side of this equation gives a rough estimate of the value of $L(p, x)$, obtained by ignoring the (relatively small) deviations of the decay of $L(p, x)$ from a true exponential decay. Disregarding the -2 in (2.5) and using the asymptotic estimate $\pi(x) \sim \frac{x}{\log x}$, (2.5) becomes $Q_5(p) \approx 1$, where

$$Q_5(p) = \frac{\pi(p)}{0.755 \log^2 S(p) - 0.755 \log S(p) \log \log S(p) - 4.19 \log S(p)}.$$

Our empirical data (cf. Figure 6) supports the validity of this approximation. Note that this figure does not exhibit the slightly increasing trend observed in Figure 3 (if the term $-4.19 \log S(p)$ is ignored then that trend becomes clearly visible). Using the rough approximation $p_k \approx k \log k$ to solve $Q_5(p) \approx 1$ in order to get p yields

$$p \sim 1.51 \log^2 S(p) \log \log S(p).$$

Remarkably, this result is consistent with the Granville conjecture with $C = C_2^{-1}$. However, this may be what happens for a typical first occurrence. Extreme values (the • points) may behave differently, perhaps in a way consistent with the Granville conjecture with $C = 2e^{-\gamma} C_2^{-1}$. As stated before, much more data is needed to settle this issue by empirical means.

TABLE 8. Record-breaking values of g_k for $p_k \leq 4 \cdot 10^{18}$.

p_k	g_k	p_k	g_k	p_k	g_k
2	1	1221 64747	222	134 62943 10749	582
3	2	1896 95659	234	140 86954 93609	588
7	4	1919 12783	248	196 81885 56461	602
23	6	3870 96133	250	261 49417 10599	652
89	8	4362 73009	282	717 71626 11713	674
113	14	12942 68491	288	1382 90485 59701	716
523	18	14531 68141	292	1958 13341 92423	766
887	20	23009 42549	320	4284 22839 25351	778
1129	22	38426 10773	336	9087 43294 11493	804
1327	34	43024 07359	354	17123 13424 20521	806
9551	36	1 07269 04659	382	21820 94054 36543	906
15683	44	2 06780 48297	384	1 18945 99698 25483	916
19609	52	2 23670 84959	394	1 68699 49409 55803	924
31397	72	2 50560 82087	456	1 69318 23187 46371	1132
1 55921	86	4 26526 18343	464	43 84154 78455 41059	1184
3 60653	96	12 79763 34671	468	55 35077 64319 03243	1198
3 70261	112	18 22268 96239	474	80 87362 46272 34849	1220
4 92113	114	24 11606 24143	486	203 98647 85174 55989	1224
13 49533	118	29 75010 75799	490	218 03472 11942 14273	1248
13 57201	132	30 33714 55241	500	305 40582 65210 87869	1272
20 10733	148	30 45995 08537	514	352 52122 34513 64323	1328
46 52353	154	41 66086 95821	516	401 42992 59991 53707	1356
170 51707	180	46 16905 10011	532	418 03264 59367 12127	1370
208 31323	210	61 44874 53523	534	804 21283 06866 77669	1442
473 26693	220	73 88329 27927	540	1425 17282 44376 99411	1476

2.2. **Prime gaps (and counts of twin primes).** Let $g_k = p_{k+1} - p_k$ be the gap between the consecutive primes p_k and p_{k+1}, and, for g restricted to be either 1 or a positive even integer, let $P(g)$ be the smallest prime p_k such that $g_k = g$, if one exists, of infinity otherwise. The Polignac conjecture [36] asserts that $P(g)$ is always finite. Also, let $N(g, x)$ be the number of solutions, with $p_{k+1} \leq x$, of the equation $g_k = g$. (The choice of counting limit, either $p_k \leq x$ or $p_{k+1} \leq x$, is a matter of implementation; we chose the latter because it does not require the computation of the smallest prime larger than x.)

Table 8 presents the record-breaking values of g_k, i.e., values of g_k larger than those for all smaller values of k (called maximal prime gaps), and Table 9 presents the record-breaking values of $P(g)$, that were found up to $4 \cdot 10^{18}$. To save some space, we do not present other first occurrences of prime gaps. For $p_k < 5 \cdot 10^{16}$, the previous published record of computation of prime gaps, they can be found in [31, 32, 50], were references to even earlier computations can be found (the rest can be found either on the first author's web pages or on Thomas Nicely's web pages). The entries for $g_k = 1172$, $g_k = 1186$, $g_k = 1356$ and $g_k = 1370$ were first discovered by Donald Knuth, and the entry for $g_k = 1048$ was first discovered by Bertil Nyman, in unrelated computations.

Table 9. Record-breaking values of $P(g)$ for $P(g) \leq 4 \cdot 10^{18}$.

g	$P(g)$	g	$P(g)$	g	$P(g)$
1	2	256	18728 51947	708	14367 94957 84681
2	3	264	23578 81993	722	21835 68728 45927
4	7	278	42609 28601	752	25529 45938 22687
6	23	294	56926 30189	764	32381 14816 25339
8	89	298	86505 24583	768	42368 30305 75549
10	139	314	89484 18749	774	46978 91428 49483
12	199	316	1 21091 72293	780	47191 16993 84963
16	1831	328	1 30868 61181	782	72650 72235 59111
26	2477	334	3 08271 38509	796	1 27130 98386 31957
28	2971	362	3 58777 24601	812	1 71027 09585 51941
30	4297	368	5 14305 18413	848	2 53707 06528 96083
32	5591	370	5 99423 58571	866	2 75931 76844 46707
36	9551	388	15 67987 92223	882	3 37105 54523 81147
38	30593	422	28 09748 65361	886	4 12707 41657 53081
46	81463	436	36 74590 59871	898	4 19816 81494 92463
56	82073	442	41 74705 54687	922	4 28612 92018 82221
64	89689	452	46 68551 87471	926	6 38194 41364 89827
66	1 62143	466	56 58556 95631	928	10 24431 62284 69423
70	1 73359	470	68 17532 56133	932	10 67648 05159 67939
74	4 04597	472	86 52447 09607	968	19 12499 02449 92669
80	5 42603	482	105 16027 87181	980	19 40368 49017 55939
88	5 44279	488	127 53631 52099	986	34 84747 41189 74633
92	9 27869	506	133 93477 50707	1006	37 34319 22965 58573
94	11 00977	508	184 10864 84491	1018	37 96724 08364 35909
102	14 44309	510	220 90169 10131	1040	46 24684 83928 75127
108	22 38823	518	229 64970 58133	1048	88 08967 23316 29091
116	58 45193	520	233 61672 62449	1052	89 21924 28734 19107
124	67 52623	536	537 12842 17763	1066	98 43614 75403 71287
134	69 58667	568	601 03305 72331	1094	139 03365 64467 25643
140	76 21259	576	881 77920 98461	1114	198 88751 28069 88729
142	103 43761	580	938 30813 40541	1124	203 15341 65230 88323
144	119 81443	590	2076 12522 61751	1144	236 55290 66620 07587
150	136 26257	608	2076 73305 30329	1150	293 46416 14651 35373
156	179 83717	624	2492 30339 18059	1172	400 24093 47413 22419
158	492 69581	626	3360 54804 00197	1186	404 44469 23233 76357
166	837 51121	628	3414 00476 13391	1192	703 39072 49524 90921
186	1476 84137	632	4567 86858 80759	1202	819 61534 49961 14321
194	1667 26367	646	5102 71604 68351	1208	1331 71124 79690 25019
200	3780 43979	654	5491 60860 07427	1264	1798 55672 01943 08703
224	4098 66323	656	6586 29660 31241	1290	2980 70756 30312 38363
226	5196 53371	676	7861 08331 15261	1306	3278 01806 91024 80227
228	8958 58039	680	8238 54353 31119	1346	$> 4 \cdot 10^{18}$
254	12024 42089	688	11052 66702 35599		

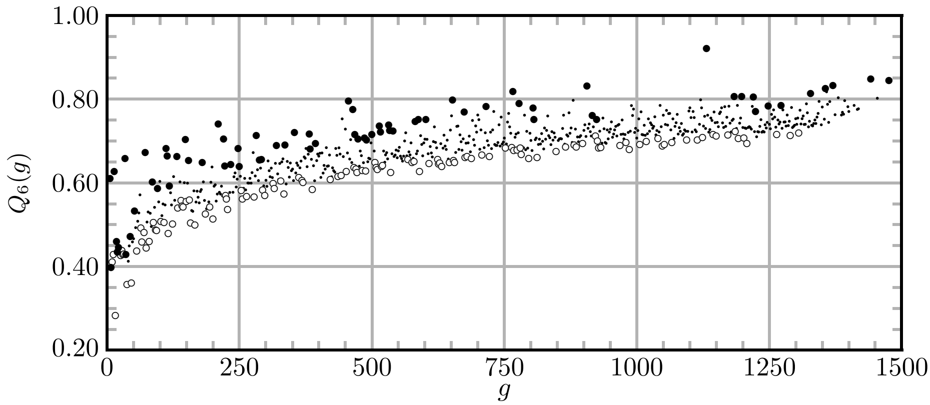

FIGURE 7. Plot of $Q_6(g)$ for $P(g) \leq 4 \cdot 10^{18}$ and for $g > 4$. Disks (•), circles (○), and dots (·) correspond respectively to data obtained from Table 8, from Table 9, and to values of $P(g)$ that did not make it to either of the two tables.

2.2.1. *Conjectures concerning prime gap upper bounds.* Cramér [7] conjectured that the equation $g > c \log^2 P(g)$ has only a finite number of solutions for $c > 1$, and an infinite number of solutions for $c < 1$, i.e., he conjectured that the largest gap between consecutive primes smaller than x should be approximately $\log^2 x$. Granville [18] conjectured that it should be $2e^{-\gamma} \log^2 x$. Shanks, on the other hand, conjectured in [43] that $g \sim \log^2 P(g)$ should hold for all first occurrences, and not only for a subsequence of them. To test these conjectures, Figure 7 presents a plot of almost all the values of

$$Q_6(g) = \frac{g}{\log^2 P(g)}$$

that we were able to compute (the points corresponding to $Q_6(1) \approx 2.08137$, to $Q_6(2) \approx 1.65707$ and to $Q_6(4) \approx 1.05637$ were omitted to reduce significantly the vertical range of the plot). Figure 7 shows that $Q_6(g)$ stays below 1 for $g > 4$ and for $P(g) < 4 \cdot 10^{18}$ (thus, also below $2e^{-\gamma} \approx 1.12292$), and that $Q_6(g)$ is slowly increasing. As explained later in subsubsection 2.2.3 the increase of $Q_6(g)$ will likely not persist for ever. Given the absence of a clear limiting value (or accumulation point) in Figure 7, our direct evidence, based solely on the first occurrence of prime gaps, is clearly insufficient to settle any of the three conjectures. As in subsubsection 2.1.1, much more data is needed before some tentative conclusions can be drawn.

2.2.2. *Estimate of $N(g, x)$ using the prime k-tuple conjecture.* From the inclusion-exclusion principle it follows that (for g positive and even)

$$N(g, x) = \sum_{\boldsymbol{s}} (-1)^{|\boldsymbol{s}|} \pi(x; \boldsymbol{s}),$$

where the sum is over all subsets $\boldsymbol{s}$ of $\{\, 0, -2, -4, \ldots, -g \,\}$ which contain 0 and $-g$. Using the prime k-tuple conjecture to approximate $\pi(x; \boldsymbol{s})$ yields

$$\hat{N}(g, x) = \sum_{k=2}^{1+g/2} (-1)^k A_{g,k} \int_c^x \frac{dt}{\log^k t}, \tag{2.6}$$

TABLE 10. Non-zero values of $A_{210,k}$ (only 21 significant digits shown).

k	$A_{210,k}$	k	$A_{210,k}$
2	$4.22503\,56214\,19965\,27314$	24	$1.30654\,90389\,76895\,22546 \cdot 10^{26}$
3	$8.55271\,41397\,87032\,74328 \cdot 10^{2}$	25	$3.98015\,77849\,08386\,50567 \cdot 10^{26}$
4	$8.36792\,68833\,33357\,58482 \cdot 10^{4}$	26	$1.07941\,25739\,42675\,11873 \cdot 10^{27}$
5	$5.27139\,32786\,77592\,64771 \cdot 10^{6}$	27	$2.60080\,37854\,40621\,31982 \cdot 10^{27}$
6	$2.40311\,09723\,02572\,10228 \cdot 10^{8}$	28	$5.55372\,35291\,47820\,70330 \cdot 10^{27}$
7	$8.44821\,97025\,17459\,94316 \cdot 10^{9}$	29	$1.04795\,36474\,29832\,87798 \cdot 10^{28}$
8	$2.38329\,96966\,57191\,74741 \cdot 10^{11}$	30	$1.74142\,93763\,38787\,42144 \cdot 10^{28}$
9	$5.54337\,34738\,69664\,85470 \cdot 10^{12}$	31	$2.53865\,33161\,01092\,25766 \cdot 10^{28}$
10	$1.08393\,95312\,84895\,97964 \cdot 10^{14}$	32	$3.23275\,89373\,30916\,84257 \cdot 10^{28}$
11	$1.80792\,42248\,15396\,08373 \cdot 10^{15}$	33	$3.57913\,90799\,32642\,56033 \cdot 10^{28}$
12	$2.60095\,23110\,19640\,17470 \cdot 10^{16}$	34	$3.42783\,40356\,80761\,84324 \cdot 10^{28}$
13	$3.25558\,92220\,22344\,78432 \cdot 10^{17}$	35	$2.82441\,80085\,26862\,50480 \cdot 10^{28}$
14	$3.56978\,17581\,63630\,82201 \cdot 10^{18}$	36	$1.99018\,32570\,50074\,25081 \cdot 10^{28}$
15	$3.44762\,24282\,49207\,49866 \cdot 10^{19}$	37	$1.19070\,12781\,96056\,59918 \cdot 10^{28}$
16	$2.94524\,14940\,75784\,28189 \cdot 10^{20}$	38	$5.99032\,61021\,74504\,60492 \cdot 10^{27}$
17	$2.23304\,45335\,51780\,41017 \cdot 10^{21}$	39	$2.49657\,02568\,80552\,25160 \cdot 10^{27}$
18	$1.50646\,92038\,79818\,67663 \cdot 10^{22}$	40	$8.40819\,41558\,71382\,32490 \cdot 10^{26}$
19	$9.05996\,67381\,18660\,00136 \cdot 10^{22}$	41	$2.19451\,40146\,49314\,36474 \cdot 10^{26}$
20	$4.86355\,36308\,62522\,36983 \cdot 10^{23}$	42	$4.13354\,37049\,56213\,13673 \cdot 10^{25}$
21	$2.33219\,01487\,92830\,32932 \cdot 10^{24}$	43	$4.93576\,18160\,53210\,32685 \cdot 10^{24}$
22	$9.99223\,09979\,82591\,31946 \cdot 10^{24}$	44	$2.76114\,18521\,61063\,83771 \cdot 10^{23}$
23	$3.82427\,45568\,44084\,48541 \cdot 10^{25}$		

where $A_{g,k} = \sum_{|\boldsymbol{s}|=k} G(\boldsymbol{s})$ and where $G(\boldsymbol{s})$ is given by (2.3). The $A_{g,k}$ constants can be computed using the method described in [6] (our $A_{g,k}$ constants are equal to Brent's $(-1)^k A_{r,k-1}$ constants, where $g = 2r$). The second author computed them all for $g \le 212$ using about 40 one-core CPU years (the first author double-checked the results for $g \le 190$). As an example of the general behavior of these constants, Table 10 presents the non-zero values of $A_{210,k}$.

Just like in subsubsection 2.1.2, it turns out that the lower limit of integration of 2 is also a very bad choice for (2.6); both $c = 0$ and $c = g$ give very good approximations to $N(g,x)$ (remarkably, $\bigl|\hat{N}(g,g)\bigr| < 6$ for $g \le 212$). In all of our comparisons between $N(g,x)$ and $\hat{N}(g,x)$ a lower limit of integration of $c = 0$ was used. Truncated versions of (2.6) behaved just like the truncated versions of (2.4) did: good approximations require all or, for small x, almost all terms.

As before, it seems reasonable to apply the law of the iterated logarithm to attempt to bound $\bigl|\hat{N}(g,x) - N(g,x)\bigr|$ by $\sqrt{2N(g,x)\log\log N(g,x)}$. To test the accuracy of this error bound estimate, Figure 8 plots some values of

$$Q_7(g,x) = \frac{\hat{N}(g,x) - N(g,x)}{\sqrt{2N(g,x)\log\log N(g,x)}}.$$

Like $Q_2(p,x)$, it appears that $\bigl|Q_7(g,x)\bigr|$ may indeed be bounded. In this case the factor of two inside the square root appears to be about right. Given that $\hat{N}(g,x) \sim A_{g,2}\frac{x}{\log^2 x}$, we should have $N(g,x) = O\bigl(\frac{x}{\log^2 x}\bigr)$, and so our empirical

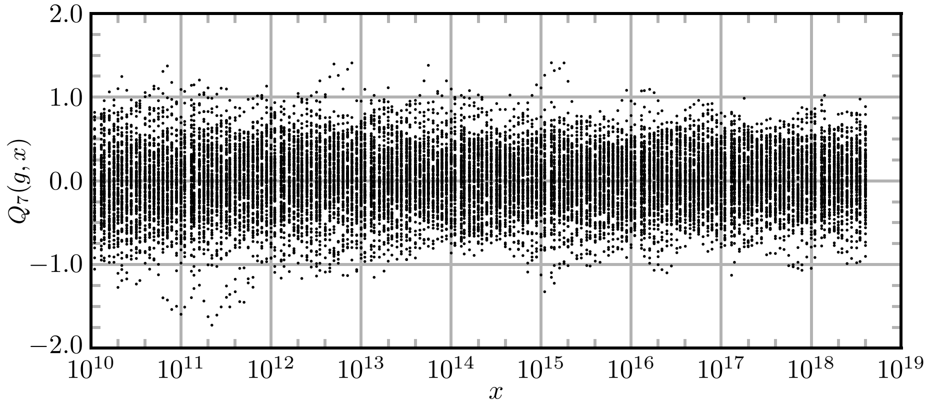

FIGURE 8. Plot of $Q_7(x)$ for $2 \le g \le 212$ and for some values of x.

data suggests that, asymptotically, one should have

$$\left|\hat{N}(g,x) - N(g,x)\right| = O\left(\frac{\sqrt{x \log\log x}}{\log x}\right),$$

where now the constant implied by the O notation depends on g. It may very well be that a similar result, with appropriate modifications, holds for the prime k-tuple conjecture itself. Numerical experiments up to 10^{17} appear to confirm that this is so.

2.2.3. *Rate of decay of* $N(g,x)$. It appears that, on a logarithmic scale, $N(g,x)$ does not deviate much from $A_{g,2}\int_0^x \frac{dt}{\log^2 t}\exp\bigl(-g/(0.960\log x - 3.58)\bigr)$ (see, for example, Figure 1 of [33] or Figure 2 of [49]). This empirical result was obtained using a method similar to that used in subsubsection 2.1.3 to quantify the decay rate of $L(p,x)$. According to [33, 49] the exponent should be, asymptotically, $-g/\log x$, which agrees reasonably well with our empirical results. The more prominent deviations from a true exponential behavior are, in this case, due to the multiplicative factors $A_{g,2} = 2C_2\prod_{p|g}\frac{p-1}{p-2}$ that are associated with the main term of $\hat{N}(g,x)$. To study the residual deviation of the exponential decay of $N(g,x)$, Figure 9 presents a plot of some values of

$$Q_8(g) = \frac{1}{A_{g,2}}\,5\cdot 10^{-16}e^{0.0266g}N(g, 4\cdot 10^{18}).$$

The factor $e^{0.0266g}$ removes most of the exponential decay of $N(g, 4\cdot 10^{18})$. The scale factor $5\cdot 10^{-16} \approx \log^2 4\cdot 10^{18}/4\cdot 10^{18}$ places $Q_8(p)$ close to 1. Similar behavior was observed for other values of x (with different exponents and scale factors). We were unable to explain the residual pattern observed in Figure 9.

Just like what was done in subsubsection 2.1.3 to estimate the order of magnitude of $S(p)$, the order of magnitude of $P(g)$ (or the order of magnitude of the largest g for a given x) can be estimated by solving

$$\frac{2x}{\log^2 x}\exp\left(-\frac{g}{0.960\log x - 3.58}\right) = 1.$$

The left-hand side of this equation gives a rough estimate of the value of $N(g,x)$, obtained by ignoring the (relatively small) deviations of the decay of $N(g,x)$ from

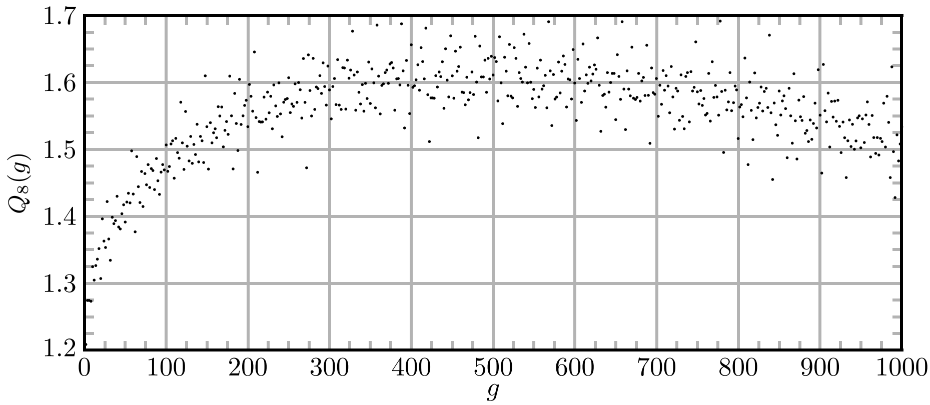

FIGURE 9. Plot of $Q_8(g)$, for $2 \le g \le 1000$.

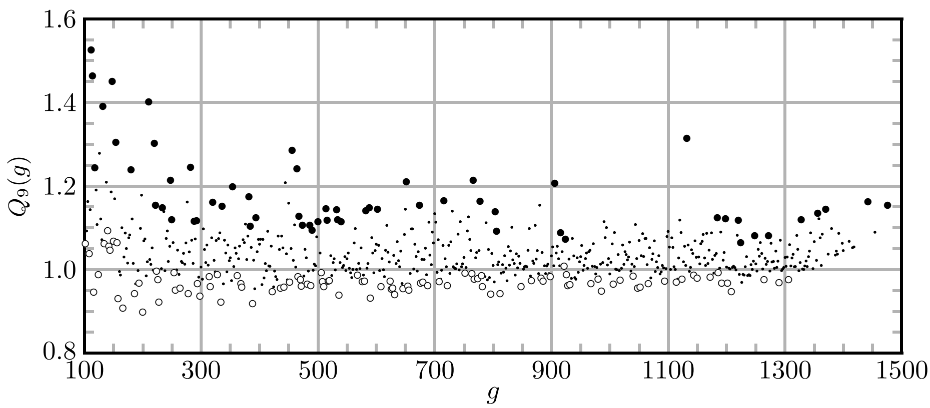

FIGURE 10. Plot of $Q_9(g)$, for $g \ge 100$ and $P(g) < 4 \cdot 10^{18}$. Disks (•), circles (∘), and dots (·) correspond respectively to data obtained from Table 8, from Table 9, and to values of $P(g)$ that did not make it to either of the two tables.

a true exponential decay and by replacing $A_{g,2}$ by its average value of 2. We get $Q_9(g) \approx 1$, where

$$Q_9(g) = \frac{g}{(0.960 \log P(g) - 3.58)(\log P(g) - 2 \log\log P(g) + \log 2)}.$$

Our empirical data (cf. Figure 10) supports the validity of this approximation. The absence of the term $-3.58 \log P(g)$ in the denominator of $Q_6(g)$ appears to be responsible for most of the increasing trend observed in Figure 7. Remarkably, $Q_9(g) \approx 1$ gives $g \sim 0.96 \log^2 P(g)$, which is close to Shanks' conjecture. It may be that typical first occurrences behave as Shanks' conjecture predicts, and that maximal prime gap occurrences (the • points of Figures 7 and 10) behave as Granville predicts. As in subsubsection 2.1.3, much more data is needed to settle this issue (by empirical means).

TABLE 11. Number of twin-primes.

k	$\pi_2(10^k)$	$\pi_2(2\cdot 10^k)$	$\pi_2(4\cdot 10^k)$
12	18705 85220	35527 70943	67568 32076
13	1 58346 64872	3 01988 62775	5 76572 48284
14	13 57803 21665	25 98584 00254	49 77948 45572
15	117 72092 42304	225 97583 03674	434 14016 30211
16	1030 41956 97298	1983 18470 25792	3819 68438 33352
17	9094 88393 53159	17544 83288 23978	33867 25524 19828
18	80867 58885 77436	1 56320 34990 75902	3 02346 31232 35320

TABLE 12. Normalized prime gap moments, and corresponding best least-squares fit data.

x	$\frac{D_2(x)}{2x\log x}$	$\frac{D_3(x)}{6x\log^2 x}$	$\frac{D_4(x)}{24x\log^3 x}$
10^{10}	0.84640 98596	0.69745 79430	0.56752 97645
10^{11}	0.85853 04971	0.71959 94626	0.59635 95130
10^{12}	0.86878 26270	0.73858 95560	0.62149 28727
10^{13}	0.87758 46594	0.75507 98973	0.64360 59388
10^{14}	0.88521 89506	0.76951 03964	0.66316 24243
10^{15}	0.89190 91355	0.78225 50563	0.68059 59792
10^{16}	0.89782 13100	0.79359 38057	0.69623 28171
10^{17}	0.90308 62730	0.80375 06718	0.71033 90224
10^{18}	0.90780 65824	0.81290 43169	0.72313 23343
best fit data	$k=2$	$k=3$	$k=4$
d_{k0}	0.99260	0.98357	0.97109
d_{k1}	−3.7012	−7.6839	−11.515
d_{k2}	7.7338	25.268	51.238
$\max_x \frac{\lvert D_k(x)-\hat{D}_k(x)\rvert}{k!\,x\log^{k-1}x}$	$3.2\cdot 10^{-5}$	$6.5\cdot 10^{-5}$	$1.6\cdot 10^{-4}$

2.2.4. *Counts of twin-primes.* As usual, let $\pi_2(x)$ be the number of twin-primes up to x, i.e., let it be the number of solutions, with $p_k \le x$, of $g_k = 2$. When x is an even integer, $\pi_2(x)$ differs from $N(2,x)$ only when x lies in the middle of a twin-prime pair. Contrary to what happens to the $\pi(x)$ function, the only known way to compute $\pi_2(x)$ is to enumerate all twin-primes up to x. Table 11 presents a small subset of the values of $\pi_2(x)$ collected during our verification of the Goldbach conjecture. As expected, $\pi_2(10^{16})$ agrees with the value found by Pascal Sebah and Xavier Gourdon in their computation of an estimate of Brun's constant [42].

2.3. **Prime gap moments.** Let

$$D_k(x) = \sum_{p_{i+1}\le x} (p_{i+1}-p_i)^k$$

be the k-order prime gap moment. In 1982 Heath-Brown [23] conjectured that $D_2(x) \sim 2x\log x$. As suggested by the first author (based solely on empirical evidence), and corroborated by Heath-Brown in an email exchange in April 2011, the following more general conjecture is plausible:

$$D_k(x) \sim k!\,x\log^{k-1}x, \qquad k \ge 1$$

(the generalization to non-integral k is obvious). The upper part of Table 12 presents some empirical data supporting this conjecture. As suggested by Heath-Brown, it turns out that our empirical data is very well approximated by

$$\hat{D}_k(x) = k!\, x \log^{k-1} x \sum_{n=0}^{N} \frac{d_{kn}}{\log^n x},$$

where N is the order of the approximation. The lower part of Table 12 presents the d_{kn} coefficients, to five significant figures, obtained by performing second order ($N = 2$) best least-squares fits to the normalized data. Twenty approximately equispaced (on a logarithmic scale) data points per decade, for $10^{10} \leq x \leq 4 \cdot 10^{18}$, were used to perform these fits. The last row presents the normalized worst observed absolute error for all of these data points, obtained using full-precision coefficients. Using a higher-order approximation, or using data starting at a higher value of x, produced even better fits, with d_{k0} coefficients even closer to one (it appears that we do not have enough data to estimate reliably the remaining coefficients).

2.4. **Verification limit of the odd Goldbach conjecture.** The odd Goldbach conjecture states that every odd number larger than 5 is the sum of three prime numbers. It is known to be true for all odd numbers larger than e^{3100} [29], and for all odd numbers larger than 5 and smaller than $1.13256 \cdot 10^{22}$ [39]. It is also known to be true if the truth of the Riemann hypothesis is assumed [10]. Without further computational effort, this last limit can be extended to $8.37 \cdot 10^{26}$ using our new verification limit of the even Goldbach conjecture and the prime gaps bounds of [39], as stated in the following theorem.

Theorem 2.1. *Each odd number larger than* 5 *and smaller than*

$$2092\,67308 \times 4 \cdot 10^{18} = 8.37069232 \cdot 10^{26}$$

is the sum of three prime numbers.

Proof. Let $N_0 = 4 \cdot 10^{18}$ and let $\Delta = 2092\,67308$. From our prime gaps results up to N_0 (cf. subsection 2.2) and, in succession, from Theorems 3 and 2 of [39], it can be inferred that, up to $N_0\Delta$, the gap between consecutive primes cannot be larger than N_0. The theorem follows by observing that using the odd primes up to $N_0\Delta$ to extend the minimal Goldbach partitions of 4, 6, …, N_0, and also of $N_0 + 2 = 211 + (N_0 - 209)$ and $N_0 + 4 = 313 + (N_0 - 309)$, will necessarily create at least one way of expressing each odd number larger than 5 and smaller than $N_0\Delta$ as a sum of three primes (actually, any sufficiently dense subsequence starting with the prime 3 will do [41]). □

Acknowledgments

In addition to the authors and their institutions (in particular, the third author used the INFN-Grid infrastructure and the "SCoPE" supercomputing center of the University of Naples Federico II), the following persons and institutions donated processor cycles to the extensive computations reported in this paper (in decreasing order of importance): NICS (National Institute for Computational Sciences, Cray XT5 Kraken supercomputer, USA), Christian Kern (Germany), National Center for Supercomputing Applications (NCSA, Xeon cluster, USA), João Rodrigues, António Teixeira, Carlos Bastos, the SIAS group, Rui Costa, Armando Pinho, and

Miguel Oliveira e Silva (all from the Department of Electronics, Telecommunications, and Informatics and from IEETA, University of Aveiro, Portugal), and Laurent Desnoguès (France). Mark Deléglise (France) provided the computer program used to compute the number of primes up to x belonging to each of the primitive residue classes modulo 120.

References

[1] Ralph G. Archibald, *Goldbach's theorem*, Scripta Mathematica **3** (1935), 44–50, 153–161.

[2] A. O. L. Atkin and D. J. Bernstein, *Prime sieves using binary quadratic forms*, Math. Comp. **73** (2004), no. 246, 1023–1030 (electronic), DOI 10.1090/S0025-5718-03-01501-1. MR2031423 (2004i:11147)

[3] Carter Bays and Richard H. Hudson, *The segmented sieve of Eratosthenes and primes in arithmetic progressions to* 10^{12}, Nordisk Tidskr. Informationsbehandling (BIT) **17** (1977), no. 2, 121–127. MR0447090 (56 #5405)

[4] Jan Bohman and Carl-Erik Fröberg, *Numerical results on the Goldbach conjecture*, Nordisk Tidskr. Informationsbehandling (BIT) **15** (1975), no. 3, 239–243. MR0389814 (52 #10644)

[5] Richard P. Brent, *The first occurrence of large gaps between successive primes*, Math. Comp. **27** (1973), 959–963. MR0330021 (48 #8360)

[6] Richard P. Brent, *The distribution of small gaps between successive primes*, Math. Comp. **28** (1974), 315–324. MR0330017 (48 #8356)

[7] Harald Cramér, *On the order of magnitude of the difference between consecutive prime numbers*, Acta Arithmetica **II** (1937), 23–46.

[8] M. Deléglise and J. Rivat, *Computing* $\pi(x)$*: the Meissel, Lehmer, Lagarias, Miller, Odlyzko method*, Math. Comp. **65** (1996), no. 213, 235–245, DOI 10.1090/S0025-5718-96-00674-6. MR1322888 (96d:11139)

[9] Marc Deléglise, Pierre Dusart, and Xavier-François Roblot, *Counting primes in residue classes*, Math. Comp. **73** (2004), no. 247, 1565–1575 (electronic), DOI 10.1090/S0025-5718-04-01649-7. MR2047102 (2005a:11152)

[10] J.-M. Deshouillers, G. Effinger, H. te Riele, and D. Zinoviev, *A complete Vinogradov* 3*-primes theorem under the Riemann hypothesis*, Electron. Res. Announc. Amer. Math. Soc. **3** (1997), 99–104, DOI 10.1090/S1079-6762-97-00031-0. MR1469323 (98g:11112)

[11] J.-M. Deshouillers, H. J. J. te Riele, and Y. Saouter, *New experimental results concerning the Goldbach conjecture*, Algorithmic Number Theory: ANTS-III Proceedings (J. P. Buhler, ed.), Lecture Notes in Computer Science, vol. 1423, Springer-Verlag, Berlin / New York, 1998, pp. 204–215.

[12] Jean-Marc Deshouillers and Herman te Riele, *On the probabilistic complexity of numerically checking the binary Goldbach conjecture in certain intervals*, Number Theory and Its Applications (S. Kanemitsu and K. Gÿory, eds.), Kluwer Academic Publishers, Dordrecht / Boston / London, 1999, pp. 89–99.

[13] Leonard Eugene Dickson, *History of the theory of numbers*, vol. I: Divisibility and Primality, AMS Chelsea Publishing, Providence, Rhode Island, USA, 1992, Published originally by the Carnegie Institute of Washington (publication number 256) in 1919.

[14] Brian Dunten, Julie Jones, and Jonathan Sorenson, *A space-efficient fast prime number sieve*, Inform. Process. Lett. **59** (1996), no. 2, 79–84, DOI 10.1016/0020-0190(96)00099-3. MR1409956 (97g:11141)

[15] W. Feller, *The general form of the so-called law of the iterated logarithm*, Trans. Amer. Math. Soc. **54** (1943), 373–402. MR0009263 (5,125c)

[16] P. X. Gallagher, *On the distribution of primes in short intervals*, Mathematika **23** (1976), no. 1, 4–9. MR0409385 (53 #13140)

[17] William F. Galway, *Dissecting a sieve to cut its need for space*, Algorithmic number theory (Leiden, 2000), Lecture Notes in Comput. Sci., vol. 1838, Springer, Berlin, 2000, pp. 297–312, DOI 10.1007/10722028_17. MR1850613 (2002g:11176)

[18] A. Granville, *Harald Cramér and the distribution of prime numbers*, Scandinavian Actuarial Journal **1995** (1995), no. 1, 12–28.

[19] A. Granville, J. van de Lune, and H. J. J. te Riele, *Checking the Goldbach conjecture on a vector computer*, Number Theory and Applications (R. A. Mollin, ed.), Kluwer Academic Publishers, Dordrecht / Boston / London, 1989, pp. 423–433.

[20] Richard K. Guy, *Unsolved problems in number theory*, 3rd ed., Problem Books in Mathematics, Springer-Verlag, New York, 2004. MR2076335 (2005h:11003)

[21] G. H. Hardy and J. E. Littlewood, *Some problems of 'partitio numerorum'; III: On the expression of a number as a sum of primes*, Acta Mathematica **44** (1922), 1–70.

[22] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, 5th ed., The Clarendon Press Oxford University Press, New York, 1979. MR568909 (81i:10002)

[23] D. R. Heath-Brown, *Gaps between primes, and the pair correlation of zeros of the zeta function*, Acta Arith. **41** (1982), no. 1, 85–99. MR667711 (83m:10078)

[24] Chen Jing-Run, *On the representation of a large even number as the sum of a prime and the product of at most two primes*, Sci. Sinica **21** (1978), 157–176, In chinese.

[25] Donald E. Knuth, 2006, PRIME-SIEVE-SPARSE program, retrieved on March 2012 from `http://www-cs-faculty.stanford.edu/~uno/programs/prime-sieve-sparse.w`.

[26] Emmanuel Kowalski, *Averages of Euler products, distribution of singular series and the ubiquity of Poisson distribution*, Acta Arith. **148** (2011), no. 2, 153–187, DOI 10.4064/aa148-2-4. MR2786162 (2012d:11199)

[27] J. C. Lagarias, V. S. Miller, and A. M. Odlyzko, *Computing $\pi(x)$: the Meissel-Lehmer method*, Math. Comp. **44** (1985), no. 170, 537–560, DOI 10.2307/2007973. MR777285 (86h:11111)

[28] W. A. Light, J. Forrest, N. Hammond, and S. Roe, *A note on Goldbach's conjecture*, BIT **20** (1980), no. 4, 525, DOI 10.1007/BF01933648. MR605912 (82h:10003)

[29] Ming-Chit Liu and Tianze Wang, *On the Vinogradov bound in the three primes Goldbach conjecture*, Acta Arith. **105** (2002), no. 2, 133–175, DOI 10.4064/aa105-2-3. MR1932763 (2003i:11147)

[30] Wen Chao Lu, *Exceptional set of Goldbach number*, J. Number Theory **130** (2010), no. 10, 2359–2392, DOI 10.1016/j.jnt.2010.03.017. MR2660899 (2011f:11133)

[31] Thomas R. Nicely, *New maximal prime gaps and first occurrences*, Math. Comp. **68** (1999), no. 227, 1311–1315, DOI 10.1090/S0025-5718-99-01065-0. MR1627813 (99i:11004)

[32] Bertil Nyman and Thomas R. Nicely, *New prime gaps between* 10^{15} *and* 5×10^{16}, J. Integer Seq. **6** (2003), no. 3, Article 03.3.1, 6 pp. (electronic). MR1997838 (2004e:11143)

[33] Andrew Odlyzko, Michael Rubinstein, and Marek Wolf, *Jumping champions*, Experiment. Math. **8** (1999), no. 2, 107–118. MR1700573 (2000f:11164)

[34] Tomás Oliveira e Silva, *Fast implementation of the segmented sieve of Eratosthenes*, Available at `http://www.ieeta.pt/~tos/software/prime_sieve.html#n`, August 2003, 2010.

[35] Tomás Oliveira e Silva, *Computing $\pi(x)$: the combinatorial method*, Revista do DETUA **4** (2006), no. 6, 759–768, Available at `http://www.ieeta.pt/~tos/bib/5.4.html`.

[36] Alphonse de Polignac, *Six propositions arithmologiques déduites du cribe d'Eratosthène*, Nouvelles Annales de Mathématiques **8** (1849), 423–429.

[37] Paul Pritchard, *Explaining the wheel sieve*, Acta Inform. **17** (1982), no. 4, 477–485, DOI 10.1007/BF00264164. MR685983 (84g:10015)

[38] Paul Pritchard, *Fast compact prime number sieves (among others)*, J. Algorithms **4** (1983), no. 4, 332–344, DOI 10.1016/0196-6774(83)90014-7. MR729229 (85h:11080)

[39] Olivier Ramaré and Yannick Saouter, *Short effective intervals containing primes*, J. Number Theory **98** (2003), no. 1, 10–33, DOI 10.1016/S0022-314X(02)00029-X. MR1950435 (2004a:11095)

[40] Jörg Richstein, *Verifying the Goldbach conjecture up to* $4 \cdot 10^{14}$, Math. Comp. **70** (2001), no. 236, 1745–1749 (electronic), DOI 10.1090/S0025-5718-00-01290-4. MR1836932 (2002c:11131)

[41] Yannick Saouter, *Checking the odd Goldbach conjecture up to* 10^{20}, Math. Comp. **67** (1998), no. 222, 863–866, DOI 10.1090/S0025-5718-98-00928-4. MR1451327 (98g:11115)

[42] Pascal Sebah and Xavier Gourdon, *Introduction to twin primes and Brun's constant computation*, Retrieved from `http://numbers.computation.free.fr/Constants/Primes/twin.html` on March 2012, 2002.

[43] Daniel Shanks, *On maximal gaps between successive primes*, Math. Comp. **18** (1964), 646–651. MR0167472 (29 #4745)

[44] Mok-kong Shen, *On checking the Goldbach conjecture*, Nordisk Tidskr. Informations-Behandling **4** (1964), 243–245. MR0172834 (30 #3051)

[45] Richard C. Singleton, *Algorithm* 357*: An efficient prime number generator*, Communications of the ACM **12** (1969), no. 10, 563–564.

[46] Matti K. Sinisalo, *Checking the Goldbach conjecture up to* $4 \cdot 10^{11}$, Math. Comp. **61** (1993), no. 204, 931–934, DOI 10.2307/2153264. MR1185250 (94a:11157)

[47] M. L. Stein and P. R. Stein, *Experimental results on additive* 2*-bases*, Mathematics of Computation **19** (1965), no. 91, 427–434.

[48] Terence Tao, *Every odd number greater than* 1 *is the sum of at most five primes*, Math. Comp., published electronically June 24, 2013.

[49] Marek Wolf, *Some heuristics on the gaps between consecutive primes*, `arXiv:1102.0481v2 [math.NT]`, May 2011.

[50] Jeff Young and Aaron Potler, *First occurrence prime gaps*, Math. Comp. **52** (1989), no. 185, 221–224, DOI 10.2307/2008665. MR947470 (89f:11019)

Departamento de Electrónica, Telecomunicações e Informática / IEETA, Universidade de Aveiro, Portugal
E-mail address: tos@ua.pt
URL: http://www.ieeta.pt/~tos

Mont Alto Campus, The Pennsylvania State University, One Campus Drive, Mont Alto, Pennsylvania 17237
E-mail address: hgn@psu.edu
URL: http://mac6.ma.psu.edu

INFN–Sezione di Napoli, Italy
E-mail address: spardi@na.infn.it