New Experimental Results Concerning the Goldbach Conjecture*

J-M. Deshouillers[1], H.J.J. te Riele[2], and Y. Saouter[3]

[1] Mathématiques Stochastiques
Université Victor Segalen Bordeaux 2
F-33076 Bordeaux Cedex, France
J-M.Deshouillers@u-bordeaux2.fr

[2] CWI, Centre for Mathematics and Computer Science
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
herman@cwi.nl

[3] Institut de Recherche en Informatique de Toulouse
118 route de Narbonne, F-31062 Toulouse Cedex, France
Yannick.Saouter@irit.fr

Abstract. The Goldbach conjecture states that every even integer ≥ 4 can be written as a sum of two prime numbers. It is known to be true up to 4×10^{11}. In this paper, new experiments on a Cray C916 supercomputer and on an SGI compute server with 18 R10000 CPUs are described, which extend this bound to 10^{14}. Two consequences are that (1) under the assumption of the Generalized Riemann hypothesis, every odd number ≥ 7 can be written as a sum of three prime numbers, and (2) under the assumption of the Riemann hypothesis, every even positive integer can be written as a sum of at most four prime numbers. In addition, we have verified the Goldbach conjecture for all the even numbers in the intervals $[10^{5i}, 10^{5i}+10^8]$, for $i = 3, 4, \ldots, 20$ and $[10^{10i}, 10^{10i}+10^9]$, for $i = 20, 21, \ldots, 30$.

A heuristic model is given which predicts the average number of steps needed to verify the Goldbach conjecture on a given interval. Our experimental results are in good agreement with this prediction. This adds to the evidence of the truth of the Goldbach conjecture.

1991 Mathematics Subject Classification: Primary 11P32; Secondary 11Y99
1991 Computing Reviews Classification System: F.2.1
Keywords and Phrases: Goldbach conjecture, sum of primes, primality test, vector computer, Cray C916, cluster of workstations

Acknowledgements

The first named author benefited from the support of CNRS and the Universities Bordeaux 1 and Bordeaux 2. The second author's contribution was carried out

* To appear in the Proceedings of the Algorithmic Number Theory Symposium III (Reed College, Portland, Oregon, USA, June 21–25, 1998).

within CWI Project MAS2.5 "Computational number theory and data security". He acknowledges the help of Walter Lioen with proving primality of many large numbers with the programs of Cohen, Lenstra and Winter, and of Bosma and Van der Hulst.
Access to the Cray C916 vector computer at the Academic Computing Centre Amsterdam (SARA) was provided by the Dutch National Computing Facilities Foundation NCF. Access to the Power Challenge Array R10000 compute server was provided by the Centre Charles Hermite in Nancy, thanks to INRIA Lorraine.

1 Introduction

The *binary* Goldbach conjecture (BGC) states that every even integer ≥ 4 can be expressed as a sum of two prime numbers. By G_2 we denote the least upper bound for the number G with the property that all even numbers n with $4 \leq n \leq G$ can be written as a sum of two prime numbers. It is known that $G_2 \geq 4 \times 10^{11}$ [15, 17, 7, 16].

The *ternary* Goldbach conjecture (TGC) states that every odd integer ≥ 7 can be expressed as a sum of three prime numbers. Clearly, the truth of BGC implies the truth of TGC.

In 1923, Hardy and Littlewood [8] proved that, under the assumption of a weak version of the Generalized Riemann hypothesis (GRH), there exists a positive integer M_0 such that TGC holds for all odd integers $\geq M_0$. In 1937, Vinogradov [18] proved, unconditionally, that there exists a positive integer N_0 such that TGC holds for all odd integers $\geq N_0$.

In 1989, Chen and Wang [3] showed that one can take $N_0 = 10^{43000}$, and in 1993 [4] they showed, assuming GRH, that one can take $M_0 = 10^{50}$. Very recently, Zinoviev [19] proved, assuming GRH, that one can take $M_0 = 10^{20}$. By the use of classical computations by Schoenfeld [14], this result implies [6]

Theorem A *If GRH holds and if $G_2 \geq 1.615 \times 10^{12}$, then every odd integer ≥ 7 can be expressed as a sum of three primes.*

This was one of our motivations for the present study.

Remark In [13], the third author has proved, unconditionally, the truth of TGC up to 10^{20} by computing an increasing sequence of about 2.5×10^8 prime numbers $q_0, q_1, \ldots, q_Q$ such that $q_0 < 4 \times 10^{11}$, $q_{i+1} - q_i < 4 \times 10^{11}$ for all $0 \leq i \leq Q - 1$ and $q_Q > 10^{20}$. This shows that near every odd number $N < 10^{20}$ there is a prime q such that $N - q < 4 \times 10^{11}$ and by [16] $N - q$ can be expressed as a sum of two primes.

A second motivation was the following result of Kaniecki [10]:

Theorem B *If the Riemann hypothesis (RH) holds and if $G_2 \geq 1.405 \times 10^{12}$,* *then every even positive integer can be written as a sum of at most four primes.*

Without any assumption, Ramaré [12] proved that every even positive integer is a sum of at most six primes.

In this paper, we report the results of extensive computer experiments to the effect of the following

Theorem 1 *We have* $G_2 \geq 10^{14}$,

so the assumptions on G_2 in Theorems A and B are satisfied.

In addition, we have checked that all the even integers in some given intervals are sums of two primes, namely:

Theorem 2 *All the even integers in the intervals* $[10^{5i}, 10^{5i} + 10^8]$, *for* $i = 3, 4, \ldots, 20$ *and* $[10^{10i}, 10^{10i} + 10^9]$, *for* $i = 20, 21, \ldots, 30$, *are sums of two primes.*

We have verified BGC with an algorithm which was used, but not given very explicitly, by Mok-Kong Shen [15]. In addition to extending the interval on which BGC is known to be true by a factor of 250, we give a heuristic model which predicts the average number of steps necessary to check BGC with this algorithm. This adds some theoretical evidence to the already overwhelming numerical evidence of the truth of BGC.

2 Two algorithms to verify the binary Goldbach conjecture on $[a, b]$

The known algorithms for verifying the Goldbach conjecture on a given interval $[a, b]$ consist of finding two sets of primes $\mathcal{P}$ and $\mathcal{Q}$ such that $\mathcal{P} + \mathcal{Q}$ covers all the even numbers in $[a, b]$.

Let p_i be the i–th odd prime number. One approach, as applied in [17, 7, 16], is to find, for every even $e \in [a, b]$, the smallest odd prime p_i such that $e - p_i$ is a prime. This amounts to taking for $\mathcal{P}$ the odd primes $p_1, p_2, \ldots, p_m$ for suitable m and to take

$$\mathcal{Q} = \mathcal{Q}(a, b) = \{q \mid q \text{ prime and } a - \epsilon_a \leq q \leq b\}$$

for some suitably chosen ϵ_a. A series of sets of even numbers $\mathcal{E}_0 \subset \mathcal{E}_1 \subseteq \mathcal{E}_2 \subseteq \ldots$ is then generated, defined by $\mathcal{E}_0 = \emptyset$,

$$\mathcal{E}_{i+1} = \mathcal{E}_i \cup (\mathcal{Q}(a, b) + p_{i+1}), \quad i = 0, 1, \ldots,^1$$

until for some j the set $\mathcal{E}_j$ covers *all* the even numbers in the interval $[a, b]$. The set $\mathcal{Q}(a, b)$ is generated with the sieve of Eratosthenes: this is the most time-consuming part of the computation. For the choice of ϵ_a it is sufficient that

[1] By $\mathcal{Q}(a, b) + p_{i+1}$ we mean, as usually, the set $\{q + p_{i+1} | q \in \mathcal{Q}(a, b)\}$.

ϵ_a exceeds the largest odd prime p_j used in the generation of the sets $\mathcal{E}_j$. This approach permits to deliver, for every even integer $e \in [a,b]$, the smallest prime p such that $e-p$ is prime (the pair $(p, e-p)$ is then called the *minimal* Goldbach decomposition of e). In the computations used for checking the Goldbach conjecture up to 4×10^{11} [16], the largest *small* odd prime needed was $p_{446} = 3163$ (this is the smallest prime p for which $244,885,595,672 - p$ is prime). An expensive part of this approach is that essentially all the primes on the interval $[a,b]$ have to be determined.

A more efficient approach, as applied in [15], is to find, for every even $e \in [a,b]$, a prime q, close to a, for which $e-q$ is a prime. This amounts to choosing for $\mathcal{P}$ the set of all the odd primes up to about $b-a$ and for $\mathcal{Q}$ the k largest primes $q_1 < q_2 \ldots < q_k$ below a, for suitable k. For the actual check of the interval $[a,b]$, one generates the sets of even numbers $\mathcal{F}_0 \subset \mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \ldots$, defined by $\mathcal{F}_0 = \emptyset$,

$$\mathcal{F}_{i+1} = \mathcal{F}_i \cup (\mathcal{P} + q_{i+1}), \quad i = 0, 1, \ldots,$$

until for some j the set $\mathcal{F}_j$ covers *all* the even numbers in the interval $[a,b]$. The large set $\mathcal{P}$ is generated with the sieve of Eratosthenes, but this work has to be done only once if we *fix* the length $b-a$ of the intervals $[a,b]$. The primes in $\mathcal{Q}$ depend on a and could also be generated with the sieve of Eratosthenes. However, since we only need a few hundred of such primes and since they do not exceed 10^{14}, it is much cheaper to use results of Jaeschke [9] by which for each prime we only need to do a few pseudoprimality tests, as long as they do not exceed 3.4×10^{14}. A disadvantage of this approach is that it does *not*, in general, find the *minimal* Goldbach decomposition.

In this study we have chosen to implement the second approach. Apart from extending G_2 as much as possible, we are interested in the *number of steps* in the above algorithms, necessary to verify BGC. In the next section we discuss a heuristic model which is capable to predict the *average* number of steps accurately.

3 Predicting the average number of steps needed to verify BGC on $[a,b]$

We present some heuristics to estimate the average number of steps needed to generate the sets $\mathcal{F}_i, i = 0, 1, \ldots$ until all the even numbers in $[a,b]$ are covered.

Let $l = b - a$ be large enough, compared with a, so that we can find enough primes q in the vicinity of a for our purpose. The number of primes in $\mathcal{P}$ is about $\pi(l)$. For each prime $q \in \mathcal{Q}$, the set $\mathcal{P} + q$ covers about $\pi(l)$ elements in $[a,b]$, i.e. a proportion of about $1 - 2\pi(l)/l$ of the even numbers in $[a,b]$ is not covered. If we assume, which is not the case, a statistical independence between the fact to be covered by $\mathcal{P}+q$ and the fact to be covered by $\mathcal{P}+q'$ and a further hypothesis of uniformity, we may expect that, on average, all even integers are covered with the help of k elements q when $(1 - 2\pi(l)/l)^k$ is roughly equal to $2/l$, the inverse of the number of even numbers in $[a,b]$. If $l = 10^8$, this leads to $k \approx 145$ and for

$l = 10^9$ this yields $k \approx 187$. A more detailed study of the probabilistic model leads to a Poisson behaviour for the number of integers which are not covered; in this model, for $k \approx 148$ in the case when $l = 10^8$ (and $k \approx 191$ when $l = 10^9$) the probability to cover the whole interval $[a, b]$ is close to 1/2. However, this does not agree with our experimental observations described in the next sections. Although a sort of statistical quasi-independence seems a natural hypothesis, the uniform distribution of primes is definitely not a decent one.

A first lack of uniformity comes from the rarification of the primes (the local density of primes around x decreases when x increases). Considering only large primes, for example those between 10^7 and 10^8 to cover an interval of length $9 * 10^7$, leads to the value $k \approx 150$; this is in good agreement with the experimental mean value of the observed k's (cf. Section 5.1).

A second and more important lack of uniformity is of arithmetical nature. Let us choose a small prime r and consider the Goldbach decomposition of all the even numbers in $[a, b]$ which are coprime with $R = 3.5 \ldots r$. For each large q (prime, so coprime with R), all the primes $p \in \mathcal{P}$ which satisfy $(p + q, R) > 1$ cannot be used to represent our numbers. The number of admissible classes of primes is thus $(3-2)(5-2)\ldots(r-2)$ and the proportion of useful primes in $\mathcal{P}$ is thus $\frac{(3-2)(5-2)\ldots(r-2)}{(3-1)(5-1)\ldots(r-1)}$. So, for each prime q, the set $\mathcal{P} + q$ contains about $\frac{(3-2)(5-2)\ldots(r-2)}{(3-1)(5-1)\ldots(r-1)}\pi(l)$ different even numbers and so the proportion of our even numbers in $[a, b]$ which are covered in one step is

$$\frac{(3-2)(5-2)\ldots(r-2)}{(3-1)(5-1)\ldots(r-1)}\pi(l) \Bigg/ \left(\frac{(3-1)(5-1)\ldots(r-1)}{3.5\ldots r}\frac{l}{2}\right),$$

i.e.,

$$2 \prod_{\substack{3 \le s \le r \\ s \text{ prime}}} \left(1 - \frac{1}{(s-1)^2}\right)\frac{\pi(l)}{l} = C(r)\frac{\pi(l)}{l}.$$

By the same reasoning as above, we expect k to be close to the solution of $(1 - C(r)\frac{\pi(l)}{l})^k = \frac{2R}{\phi(R)l}$. For $r = 97$ and $l = 10^8$, this leads to $k \approx 206$ and for $l = 10^9$ we find $k \approx 270$. This agrees well with our experiments and this implies, as one may expect, that for the even numbers in $[a, b]$ which are *not* coprime with $R = 3.5 \ldots r$, it is easier in general to find a Goldbach decomposition than for those which are coprime with R. Again, if we improve this model by the Poisson probabilistic consideration and the rarification of the primes, we are led to $k \approx 214$ when $l = 10^8$, which is, here also, in good agreement with the experimental data of Section 5.1. This probabilistic reasoning will be developed in a forthcoming paper.

4 Computations which extend G_2 from 4×10^{11} to 10^{14}

We have adopted Shen's approach, described in Section 2, to extend the binary Goldbach conjecture as far as possible beyond the known bound of 4×10^{11}.

The intervals $[a, b]$ were chosen to have a length of 10^8 or 128×10^6 or 10^9. The largest possible prime one needs in the set $\mathcal{P}$ lies close to $b - q_1$. By the prime number theorem, $q_1 \approx a - k \log a$, so that $b - q_1 \approx b - a + k \log a$. As *maximum* values of k we found in our experiments that $k = 430$ was sufficient. For $a \approx 10^{14}$ this implies that the largest prime in the set $\mathcal{P}$ must have a size of at least $10^9 + 1.4 \times 10^4$ for $b - a = 10^9$. In our actual implementation we have chosen $\mathcal{P}$ to contain the odd primes up to $10^8 + 10^5$ in the case $b - a = 10^8$, and those up to $10^9 + 10^6$ in the case $b - a = 10^9$.

For the actual generation of the primes close to a we have used Jaeschkes computational results [9], stating that if a positive integer $n < 215,230,289,8747$ is a strong pseudoprime with respect to the first five primes $2, 3, 5, 7, 11$, then n is prime; corresponding bounds for the first six and seven primes are, respectively, 3,474,749,660,383 and 341,550,071,728,321.

Initially, both the second and the third author have checked the BGC up to 10^{13}, *independently*, on a Cray C916 vector computer resp. on an SGI compute server with 18 R10000 CPUs. After learning about each other's results, they decided to work together to reach the bound 10^{14}. The second author has checked the BGC on the intervals $x \times 10^{13}$ for $x = [2, 4], [6, 8], [9, 10]$ and the third author those for $x = [1, 2], [4, 6], [8, 9]$.

4.1 Experiments on the Cray C916 vector computer

The second author has implemented Shen's algorithm on a Cray C916 vector computer as follows.

With the large set of odd primes $\mathcal{P}$ we associate a long bit-array called ODD, in which each bit represents an odd number $< 10^9 + 10^6$, the bit being 1 if the corresponding odd number is prime, and 0 if it is composite. With $\mathcal{F}_i$ we associate a similar bit-array called SIEVE, having the same length as ODD. The first bit of SIEVE represents the even number $q_1 + 3$, the second bit $q_1 + 5$, and, in general, bit i represents the even number $q_1 + 2i + 1$. Initially, ODD is copied into SIEVE, making bit i of array SIEVE equal to 1 if $2i+1$ is a prime, indicating that $q_1 + 2i + 1$ can be written as sum of the two primes q_1 and $2i + 1$. Now array SIEVE represents the set $\mathcal{F}_1$. In the second step, array SIEVE is "or"-ed with a right-shifted version of array ODD, where the shift equals $(q_2 - q_1)/2$. It is easy to see that now array SIEVE represents the set $\mathcal{F}_2 = \mathcal{F}_1 \cup (\mathcal{P} + q_2)$. In general, $\mathcal{F}_{i+1}$ is generated from $\mathcal{F}_i$ by doing an "or" operation between array SIEVE and array ODD, right-shifted with shift $(q_{i+1} - q_1)/2$.

Of course, these steps can be carried out very efficiently on the Cray C916. We compressed 64 bits into one word and vectorized the "or" operations. Checking whether *all* the bits of array SIEVE have become 1 is only done when the chance of occurrence of this event has become sufficiently large (after 170 steps, in our program). As soon as the number of 0-bits has dropped below 4, the remaining "stubborn" even numbers are listed in order to "see" some intermediate output.

In one typical run, we handled 1000 consecutive intervals of length 10^9. Close to 10^{14} the time to generate 1000×430 large primes was about 5000 CPU–seconds, and the total sieving time was about $13,200$ seconds. The average (over

1000 consecutive intervals) number of steps in each run varied between 269 and 271 with standard deviation between 18 and 20. The total (low priority) CPU time used to cover the intervals $[4 \times 10^{11}, 10^{13}]$, $[2-4] \times 10^{13}$, $[6-8] \times 10^{13}$, and $[9-10] \times 10^{13}$ was approximately 75 CPU–hours for generating the large primes, and 225 CPU–hours for the sieving. The latter means that in the sieving part an average of 3.2×10^8 64-bit words per CPU-second were "or"-ed. The largest number of large primes which we needed was 413: for $e = 33,836,446,494,106$ and first prime $q_1 = 33,835,999,990,007$ it turned out that $e - q_i$ is composite for $i = 1, \ldots, 412$, and prime for $i = 413$ ($q_{413} = 33,836,000,002,499$ and $e - q_{413} = 446,491,607$).

4.2 Experiments on the SGI compute server with 18 R10000's

The algorithm as implemented by the third author on the SGI workstation is very close to the one of the Cray C916 as described in Section 4.1. Prime numbers up to 128×10^6 are represented into a binary array, that we call again ODD, of one million 64 bits long entries: the j-th bit of the i-th element of the array is equal to 1 if and only if $128 * i + 2 * j + 3$ is prime. Similarly another array of the same size, corresponding to the array SIEVE of the previous section, is used to note decomposed numbers: the j-th bit of the i-th element of this latter array is equal to 1 if and only if $128 * i + 2 * j + \mathit{seed}$ is decomposable as sum of two prime numbers, where *seed* denotes the even integer at which the phase begins.

At this point, the task of the program is to fill all the entries of SIEVE with the greatest 64 bits word i.e. $2^{64} - 1$. The program searches for the least entry i for which the value of SIEVE$[i]$ is not maximum and then searches for the least bit j of this entry not being equal to 1. Thus, the number $128 * i + 2 * j + \mathit{seed}$ has still not been written as sum of two primes. The program then searches for the least value k for which $128 * (i - k) + \mathit{seed} - 3$ and $128 * k + 2 * j + 3$ are both prime. When such a k value is found the array SIEVE beginning at the entry i can be combined with the array ODD beginning at the entry k with an or operation as previously. Having a step size of 128 in the search of prime numbers does not change the density of expected prime numbers and has the advantage of avoiding the shift of the array ODD. At last, in order to gain efficiency, the addressing of the array SIEVE was done through a chained list: this list contains only values i for which SIEVE$[i]$ is not maximal. Then after each or operation, the resulting value is compared with $2^{64} - 1$ and if there is equality, the corresponding index is removed from the chained list. Thus the size of the array decreases when time elapses and globally no useless or operation is made. The drawback is that addressing has to be done by indirect pointer redirection and this slows down the program at the beginning of the execution. Versions with and without linked chain implementation were tested on a DECSTATION 3100 with various word sizes and various sizes for the arrays ODD and SIEVE. The gain of the linked chain version appeared to be maximal for arrays with a length of 1.5×10^6 words of 32 bits, with a factor of 1.59. Later, some comparisons were made with a version with prime entries up to 10^9. The ratio of time executions was 0.82 to the benefit of the latter versions. Some other improvements were not implemented, e.g. anticipating the decompositions in the block of even numbers following the one of the current array SIEVE, when indices go out of the range of this latter array.

Typical runs consisted of checking 1350 consecutive intervals of even integers of length $128 * 10^6$ with one run on each of the 18 R10000 processors of the SGI workstation. Seven such runs were necessary to deal with an interval of 2.10^{13}. Intervals that were checked are $[10^{13}, 2.10^{13}]$, $[4.10^{13}, 6.10^{13}]$, and $[8.10^{13}, 9.10^{13}]$. A total number of 324 runs was necessary to complete this whole task. User CPU times for the various runs varied from 10 hours 33 mns for the run beginning at 9,158,401,000,000 and ending at 9,331,201,000,000, up to 17 hours 12 mns for the run from 2,937,601,000,000 up to 3,110,401,000,000. The total sequential time was 4083 hours 38 mns and so the real time, which is about 18 times smaller, was about 227 hours. Those times include the search for primes *and* the sieving. The number of prime numbers needed to verify the decomposition of $64 * 10^6$ even consecutive integers varies from 160 for the intervals beginning at 16,182,785,000,000 and 53,917,312,000,000, up to 184 for the interval beginning at 145,793,000,000. When testing on intervals of length 10^9, the average number of prime numbers grows up to 218.

5 Checking BGC near high powers of ten

Apart from extending G_2, we have also checked the binary Goldbach conjecture on intervals of length 10^8 and 10^9 near high powers of ten. The second author has checked the intervals $[10^{5i}, 10^{5i} + 10^8]$, for $i = 3, 4, \ldots, 20$, and the third author has checked the intervals $[10^{10i}, 10^{10i} + 10^9]$, for $i = 20, 21, \ldots, 30$.

5.1 The intervals $[10^{5i}, 10^{5i} + 10^8]$, for $i = 3, 4, \ldots, 20$

For each interval $[B, B + 10^8]$ the largest 300 primes $\leq B$ were generated. Here, the results of Jaeschke could not be used anymore because the numbers were too large. Instead, we first generated the 300 largest numbers $\leq B$ which pass a strong pseudo-prime test for one randomly selected base, and next we proved primality of these numbers with a program developed by H. Cohen, A.K. Lenstra, and D.T. Winter [5]: all these numbers turned out to be prime. For the set $\mathcal{P}$ we took the odd primes below $10^8 + 10^6$. The sieving technique was the same as that used on the Cray C916 for the even numbers up to 10^{14}.

A selection of the results are given in Table 1. The second column gives the value of $(q_{300} - q_1)/(299 \log 10)$ which should be close to $\log_{10} B$, according to the Prime Number Theorem. It illustrates that the local behaviour of the primes may deviate considerably from the known global behaviour. The average number of steps needed (over the 18 intervals considered) was 217, with standard deviation 23. For a *uniform* distribution of bits in array ODD (instead of the distribution induced by the primes) the average number of steps was 152, with standard deviation 9. This agrees well with the expected number of steps (214 in the case of primes and 150 in the case of uniform distribution) mentioned in Section 3.

5.2 The intervals $[10^{10i}, 10^{10i} + 10^9]$, for $i = 20, 21, \ldots, 30$

Again the SGI compute server was used to make a similar implementation. For an interval of the form $[B, B+10^9]$, as in the implementation for decompositions up to 10^{14}, the even numbers were represented as bits in an array of 7812500 64–bit words. The sieving technique was the same as previously and also used chained lists. However, because of the size of the numbers, again Jaeschke's results could not be used to establish primality. Instead of that, we passed candidate numbers through Miller-Rabin pseudo-primality tests for the bases 2, 3, 5 and 7 after a quick trial division sieve. The implementation of this phase was made with the PARI system. In a second phase we certified primality of these numbers by the Elliptic Curve Primality Prover program of François Morain [1, 11]. On one R10000 node, the CPU times for the C version of ECPP, which the third author had to his disposal, varied between 4 minutes for numbers of 200 decimal digits and 60 minutes for numbers of 300 decimal digits. As a comparison, the primality of some of these numbers was proved by Cohen, Lenstra and Winter's program [5] (for numbers up to 220 decimal digits; the average CPU-time was two minutes per number on an 180 MHZ IP32 SGI workstation) and by a program of Bosma and Van der Hulst [2] (for numbers larger than 220 decimal digits; the average CPU-time was seven minutes per number on the same 180 MHZ IP32 SGI workstation[2]). The number of prime numbers required to verify the BGC on an interval of length 10^9 was in fact nearly stable, varying from 222 up to 231 for the considered intervals with an average value of 225. Table 2 summarizes the results.

[2] The CPU-time asked by this program grows with the size of the prime number, but in a very erratic way.

Table 1 Checking the Goldbach conjecture on the intervals $[B, B + 10^8]$, for $B = 10^{15}, 10^{20}, \ldots, 10^{100}$.

Notation

$[B, B + 10^8]$: the interval on which the Goldbach conjecture is verified;

$q_1, \ldots, q_{300}$: the largest 300 consecutive primes $\leq B$, generated on an SGI workstation with a 100 MHZ IP22 processor;

T_{pr} : the sum of the CPU-times in minutes spent to generate the largest 300 strong pseudo-primes $< B$ (which pass a strong pseudo-prime test for a randomly chosen base) with the PARI package, and to prove primality with a Fortran/C code based on the Cohen-Lenstra primality proving algorithm (all the strong pseudo-primes turned out to be prime);

N_1 : the smallest positive integer such that for each even number $e \in [B, B + 10^8]$ there is an index i with $1 \leq i \leq N_1$ such that $e - q_i$ is a prime number;

W : the "worst case" in the Goldbach check of the interval $[B, B + 10^8]$, i.e., $W - q_i$ is composite for $i = 1, 2, \ldots, N_1 - 1$, but prime for $i = N_1$.

$\log_{10} B$	$\frac{q_{300} - q_1}{299 \log 10}$	$B - q_1$	$B - q_{300}$	T_{pr}	N_1	$W - B$
15	16.2	11159	11	1.9	243	87831838
20	19.8	13611	11	2.4	210	40249602
25	24.6	17063	123	3.0	240	91143618
30	31.9	21941	11	4.2	216	70421718
35	34.1	23477	23	5.7	182	84348372
40	37.2	25649	17	6.0	202	61919718
45	51.5	35469	9	8.0	283	80017866
50	45.3	31247	57	11	198	84955228
55	56.8	39183	111	16	218	88062574
60	58.9	40721	161	25	210	68370894
65	66.4	45951	269	32	193	80085838
70	72.6	50093	93	43	210	56324104
75	80.7	55779	191	53	224	31058458
80	82.6	56907	11	65	206	24403128
85	76.8	52919	27	82	203	45500944
90	95.6	65981	143	94	209	70588714
95	92.9	64011	53	122	207	88980634
100	99.0	68969	797	150	250	41229036

Table 2 Checking the Goldbach conjecture on the intervals $[B, B + 10^9]$, for $B = 10^{200}, 10^{210}, \dots, 10^{300}$.

Notation

$[B, B+10^9]$: the interval on which the Goldbach conjecture is verified;

$q_1, q_2, \dots$: the list of prime numbers needed to verify BGC on $[B,\ B+10^9]$;

N_q : the cardinality of the previous set;

T_{gen} : the time needed to sieve the interval and to generate the N_q strong pseudo-primes with the PARI package on a single node of the SGI;

T_{pp} : the total sequential time spent by ECPP to prove primality of the N_q pseudoprimes (this task was in fact distributed on all nodes of the compute server);

W : the "worst case" for the verification of BGC in the interval, i.e. $W - q_i$ is composite for $i = 1, 2, \dots, N_q - 1$ but prime for $i = N_q$.

$\log_{10} B$	N_q	$\frac{q_{N_q} - q_1}{64.N_q.\log 10}$	$B - q_1$	$q_{N_q} - B$	$W - B$
200	224	30281.7	97283	999497853	999786382
210	222	30559.2	243203	999503869	999686796
220	222	30557.9	177539	999530109	999620578
230	228	29754.9	112643	999634941	999983752
240	228	29763.3	191747	999836541	999872854
250	231	29381.5	701699	999488509	999991806
260	226	30026.2	174467	999837181	999864924
270	223	30418.1	112643	999503997	999697006
280	225	30151.7	32003	999714813	999837064
290	226	30023.9	115331	999819517	999872646
300	227	29885.3	32771	999692157	999821434

$\log_{10} B$	T_{gen}	T_{pp}
200	33 mn 52 s	14 h 20 mn 02 s
210	35 mn 56 s	39 h 20 mn 31 s
220	44 mn 47 s	48 h 45 mn 05 s
230	48 mn 43 s	62 h 57 mn 39 s
240	55 mn 32 s	72 h 20 mn 47 s
250	1 h 11 mn 38 s	91 h 10 mn 43 s
260	1 h 16 mn 42 s	104 h 25 mn 31 s
270	1 h 13 mn 02 s	120 h 35 mn 28 s
280	1 h 45 mn 09 s	98 h 31 mn 15 s
290	1 h 39 mn 24 s	177 h 37 mn 48 s
300	2 h 04 mn 44 s	219 h 54 mn 59 s

References

[1] A.O.L. Atkin and F. Morain. Elliptic curves and primality proving. *Mathematics of Computation*, 61:29–68, 1993.

[2] Wieb Bosma and Marc-Paul van der Hulst. *Primality proving with cyclotomy.* PhD thesis, University of Amsterdam, December 1990.

[3] J.R. Chen and T.Z. Wang, *On the odd Goldbach problem*, Acta Math. Sinica **32** (1989), pp. 702–718 (in Chinese).

[4] J.R. Chen and T.Z. Wang, *On odd Goldbach problem under General Riemann Hypothesis*, Science in China **36** (1993), pp. 682–691.

[5] H. Cohen and A.K. Lenstra, *Implementation of a new primality test*, Math. Comp. **48** (1987), pp. 103–121.

[6] J-M. Deshouillers, G. Effinger, H. te Riele and D. Zinoviev, *A complete Vinogradov 3-primes theorem under the Riemann hypothesis*, Electronic Research Announcements of the AMS **3** (1997), pp. 99–104 (September 17, 1997); `http://www.ams.org/journals/era/home-1997.html` .

[7] 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, Dordrecht, 1989, pp. 423–433.

[8] G.H. Hardy and L.E. Littlewood, *Some problems of 'Partitio Numerorum'; III: On the expression of a number as a sum of primes*, Acta Math. **44** (1922/3), pp. 1–70.

[9] G. Jaeschke, *On strong pseudoprimes to several bases*, Math. Comp. **61** (1993), pp. 915–926.

[10] L. Kaniecki, *On Šnirelman's constant under the Riemann hypothesis*, Acta. Arithm. **72** (1995), pp. 361–374.

[11] François Morain. *Courbes Elliptiques et Tests de Primalité.* PhD thesis, L'Université Claude Bernard, Lyon I, September 1990. Introduction in French, body in English.

[12] O. Ramaré, *On Šnirel'man's Constant*, Ann. Scuola Norm. Sup. Pisa **22** (1995), pp. 645–706.

[13] Yannick Saouter, *Checking the odd Goldbach conjecture up to* 10^{20}, Math. Comp., **67** (1998), pp. 863–866.

[14] L. Schoenfeld, *Sharper Bounds for the Chebyshev Functions* $\theta(x)$ *and* $\psi(x)$. *II*, Math. Comp. **30** (1976), pp. 337–360.

[15] Mok-Kong Shen, *On Checking the Goldbach conjecture*, BIT 4 (1964), pp. 243–245.

[16] M.K. Sinisalo, *Checking the Goldbach conjecture up to* $4 \cdot 10^{11}$, Math. Comp. **61** (1993), pp. 931–934.

[17] M.L. Stein and P.R. Stein, *Experimental results on additive 2 bases*, Math. Comp. **19** (1965), pp. 427–434.

[18] I.M. Vinogradov, *Representation of an odd number as a sum of three primes*, Comptes Rendues (Doklady) de l'Académie des Sciences de l'URSS, **15** (1937), pp. 291–294.

[19] D. Zinoviev, *On Vinogradov's constant in Goldbach's ternary problem*, J. Number Th. **65** (1997), pp. 334–358.