Enumeration to 1e14 of the twin primes and Brun's constant

Thomas R. Nicely

http://www.trnicely.net
Current e-mail address

Document History

Except for the addendum (not part of the submission for publication), the content of this document is essentially that of the journal article as published; it may contain minor revisions and corrections (such as updated URLs), and differ in format and detail.

Abstract

The count $pi\_2(x)$ of the number of twin-prime pairs $<= x$, as well as the count $pi(x)$ of the number of primes $<= x$, was tabulated by means of a computer search for values of x up to 1e14, at intervals of 1e9. The floating point sum of the reciprocals was also calculated, resulting in an improved estimate for Brun's constant of 1.9021605778, with a standard deviation of 0.00000 00021. An algorithm for estimating the standard deviation of such calculations is described. In the course of the computation a flaw was discovered in the hardware divider of the floating point unit of Intel Corporation's Pentium microprocessor.

Mathematics Subject Classification 2000 (MSC2000)

Primary: 11A41.
Secondary: 11-04, 11N36, 11Y60, 11Y70, 11Y35, 68M15.

Key words and phrases

Twin primes, Brun's constant, primes, Pentium flaw, computer errors.

Introduction

The set $\{(3, 5), (5, 7), (11, 13), ...\}$ of twin-prime pairs $(q, q+2)$ has been studied by Brun (1919), Hardy and Littlewood (1923), Selmer (1942), Fröberg (1961), Weintraub (1973), Bohman (1973), Shanks and Wrench (1974), and Brent (1975, 1976). Currently M. Kutrib and J. Richstein (1996, 1997) are completing a study similar to the present one.

Although it is still not known if the set is infinite, Brun (1919) proved that in any case the sum of the reciprocals

(1)   B = (1/3 + 1/5) + (1/5 + 1/7) + (1/11 + 1/13) + (1/17 + 1/19) + ...

is convergent, in contrast to the divergent sum of the reciprocals of the individual primes; thus B is known as Brun's constant. Also unlike the individual primes, no indirect method is known for enumerating the twins; my investigation, like the others, simply sieves the individual primes to the desired upper bound, then checks directly for the twin pairs among them. The "Prime k-tuples conjecture," formulated by Hardy and Littlewood (1923), has as a corollary the following asymptotic formula for the count of twin primes:

(2) $pi\_2(x) \sim L\_2(x) = 2c\_2*int(1/(ln(t))^2, t, 2, x)$

where $pi\_2(x)$ is the number of twin-prime pairs $(q, q+2)$ such that $q <= x$, and c_2 is the "twin-primes constant,"

(3) $c\_2 = 0.66016\ 18158\ 46869\ 57392\ 78121\ ...,$

computed to 42 decimals by J. W. Wrench (1961). See Riesel (1994, pp. 60-68) for a discussion of (2), often styled the "Hardy-Littlewood approximation," and (3).

Although (1) is convergent, the partial sums approach the limit with agonizing slowness. However, assuming the Hardy-Littlewood approximation (2), an accurate estimate can easily be derived (Fröberg, 1961) for the remainder term, producing a result which converges to the same limit much more rapidly:

(4) $B*(x) = B(x) + 4c\_2/\ln(x)$,

where $B*(x)$ is the accelerated approximation to Brun's constant and $B(x)$ is the partial sum of the reciprocals of the twins:

(5) $B(x) = \text{sum}(1/q + 1/(q+2), q, q <= x)$.

Furthermore, Brent (1975) conjectures a corresponding standard deviation of

(6) $\text{sigma\_}B(x) = \text{sqrt}(8c\_2)/(\text{sqrt}(x)*\ln(x))$

for the estimates of B from $B*(x)$.

Brent (1975, 1976) extended the previous enumerations of the twins, and their reciprocal sums, to 1e11; his resulting estimate for Brun's constant was 1.90216 054 +/- 0.00000 050. To improve on this result, the author has extended these calculations three orders of magnitude, to an upper bound of 1e14.

Computational Technique; the Pentium Flaw

The calculations were carried out exclusively on personal computers powered by Intel x86 CPUs, a mixture of 486DX-33 and Pentium systems. The operating environment was Microsoft's DOS and Windows 3.x with Win32s support added. The code was written in C, using the Windows 3.x and Win32s APIs, and compiled using Borland C++ 4.02. The computational power of the individual units was enhanced by distributing the calculations independently and asynchronously across multiple systems---five 486DX-33s early on, eventually expanding to more than a dozen Pentium systems.

The most critical section of the code lies in the procedures which sieve large blocks of integers (typically twenty million, though block size is adjusted to available physical system memory) for primes. A variation of the sieve of Eratosthenes is used, with a number of tricks added to improve speed. One array holds the base primes (those <= sqrt(x)) used as trial divisors; actually, it contains the differences between successive primes, which require less storage (one byte for each prime < 4.3e8, or < 3.04e11 if half the difference is stored; see Brent (1973, p. 961) and Riesel (1994, p. 80)) and from which successive base primes are reconstructed by integer addition. The second array, usually much larger, has one byte corresponding to each odd integer in the block to be sieved; the byte is set to 1 for primes and zero for composites. The base primes are read in at startup from a disk file, which uses bit encoding (mod 30) to store primality information for 30 numbers per byte (bit encoding proved inefficient for the arrays in memory, due primarily

to the absence in Intel x86 processors of any native mode or instructions supporting bit-based memory addressing).

The cumulative effect of evolutionary optimization illustrates the incredible power residing within current personal computers. Throughput was initially about 1.7e8 integers per day on a 386/387; current versions do about 1e11 per day on modestly equipped Pentiums. This compares quite favorably with the throughput of 6.6e11 per day implied by Young and Potler (1989), employing a Cray-2 on a similar problem which required no floating point arithmetic. The biggest speed boosts came from the moves to 16-bit and 32-bit Windows, the first allowing access to all of extended memory for the arrays of primes, the second eliminating segment arithmetic.

At first calculations were performed in runs of 1e11 per machine; more recently, in runs of 1e12. Output was dumped to a binary disk file at intervals of 1e9; at the end of a run, a copy of the output file was appended to a master output file on my home system. A viewer code was written to allow analysis of the output.

Error detection and prevention was a major consideration from the outset, particularly since personal computers do not have the reputation for reliability enjoyed by workstations, mainframes, and supercomputers. A count of primes was maintained, and compared with published values (Riesel, 1994, pp. 380-383) of pi(x) at intervals. Values of pi_2(x) and the partial sums B(x) of the reciprocals were compared with those obtained by Brent (1975, 1976) to 1e11. The reciprocal sums were computed by two independent methods. One used the IEEE floating point arithmetic native to the Intel numeric coprocessors, with a 64-bit normalized mantissa (extended precision, or long double data type, giving 19 significant digits); the other calculated the reciprocals and sums using a modification of an ultra-precision integer arithmetic package graciously donated to the public domain by Arjen Lenstra (1988-1991), and as implemented produced 53 decimals (initially 26 decimals).

The reliance on floating point arithmetic was even more pervasive than it would appear, since the magnitude of the integers being investigated exceeded the capacity of the 32-bit registers native to the Intel processors. Thus integers exceeding $2^{32} - 1$ had to be represented as long double floating point values, with care required to guard against any associated rounding or truncation errors.

Naturally, many errors were encountered during early development, the result of logical flaws in the algorithms and code. Once these were eliminated, the calculations proceeded smoothly. Suddenly, after a group of runs was assembled and analyzed on 13 June 1994, a discrepancy appeared: the count of primes < 2e13 was incorrect. In the process of attempting to eliminate this error (and at the same time port the code to 32-bit Windows), it

was discovered that bugs in the Borland compiler were also producing erroneous results (due to incorrect rounding of quotients) in the ultra-precision sums. Not until 10 September 1994 was a ported version of the code produced with all known sources of error eliminated.

Ever more cautious (paranoid?), this time I restarted the entire calculation from zero and performed each run (of size 1e12) in duplicate on two machines. The first run of the first 1e12 was performed on the only Pentium in the group; it had been added to the mix the previous March. The duplicate run was completed on my wife's 486DX-33 on 4 October 1994. It was immediately clear that their ultra-precision reciprocal sums differed. Using binary search, the difference was soon isolated to the results for the twin pair 824633702442 +/- 1, for which not only the ultra-precision results differed, but even the floating point reciprocals returned from the CPU were in error. Several days were then spent looking for the culprit; compiler error, memory error, system bus, etc. By 19 October 1994 I was all but certain the error was within the floating point hardware unit of the Pentium CPU itself. After several more days fruitlessly quizzing tech support at the system vendor and finally at Intel itself, I dispatched (30 October 1994) an e-mail query regarding the error to several parties whom I thought would be interested and would have access to a large variety of systems. The message and its consequences were spread worldwide over the Internet within days, and eventually Intel admitted that such errors (which occurred only in floating point division and remaindering operations, with unusual denominators) were the result of a production flaw in nearly all (over a million) of the Pentium CPUs produced to that time. Although the error was quite rare and often of extremely small magnitude, a crisis in public relations and consumer confidence ensued, and Intel eventually agreed to replace all such chips with corrected versions, at the company's expense. In January 1995, Intel announced (*PC Week*, 1995) a $475 million accounting charge to cover the cost of the episode. More elaborate technical details may be found in Sharangpani and Barton (1994) and in Coe (1995). One final point of interest in this regard is that the original error of 13 June 1994, the incorrect count of primes, appears to have originated in the use of the fmod remaindering instruction in the sieving process; fmod in turn invokes one of the fdiv family of instructions, which is then executed in the FPU hardware divider.

After this tempest began to settle (mathematicians in general, and this one in particular, are not accustomed to being besieged by inquiries and visits from network news, the BBC, the Washington *Post*, the *Wall Street Journal*, and the New York *Times*, as well as journalists from France, Australia, and Malaysia), attention returned to the work at hand. With commendable haste, Intel provided a replacement chip for the errant Pentium (as well as one for a colleague's system) and also supplied a 90-MHz Pentium system to help make up for lost time. Naturally, all remaining calculations were carried out in duplicate, and the

wisdom of this caution was confirmed when other discrepancies appeared between duplicate runs. Twice the errors were tracked to intermittently defective memory (SIMM) chips; parity checking had failed to report either error. On another occasion, a disk subsystem failure generated a wholesale lot of incorrect, yet plausible data. In the most recent instance, a soft memory error appears to be the culprit.

Although many may dismiss machine errors such as these as of no practical consequence (except to mathematicians) or as evidence of the unreliability of personal computers, I believe there is a more important conclusion to be drawn. Any significant computation, for which there is no simple check available, should be performed in duplicate, preferably on two different systems, using different CPUs, hardware, operating systems, application software, and algorithms. In practice, it is difficult to obtain this degree of independence; but certainly, one must be suspicious of the result of any single, lengthy machine computation, whether it is performed on a personal computer or the world's finest supercomputer. As we tell our students---check your work!

Computational Results

Table 1 contains a brief summary of the computational results, including the counts $pi\_2(x)$ of twin-prime pairs; the values of Brent's (1975) $r\_3(x)$,

(7) $r\_3(x) = L\_2(x) - pi\_2(x)$;

the partial sums $B(x)$ of (5); and the extrapolations $B*(x)$ of $B(x)$ to the limit according to (4), yielding a sequence of approximations to Brun's constant believed to be of ever increasing accuracy. Note that in the interest of simplicity, both Table 1 and Table 2 use the floating point format (common to most programming languages) for numbers in scientific notation. More extensive versions of Table 1 are available at http://www.trnicely.net/twins/t2_0000.htm, at http://www.trnicely.net/twins/t2_0001.htm, and at this site. Furthermore, the latest updated count of the twin primes, and the corresponding estimates for Brun's constant and the error bound, are available at http://www.trnicely.net/counts.html.

```
 Table 1. Counts of twin-prime pairs and estimates of Brun's constant.
```

x	pi_2(x)	r_3(x)	B(x)	B*(x)
8e10	182855913	-984.74	1.796977508288414	1.902160400157630
9e10	203710414	-6872.36	1.797468808649461	1.902160532806004
1e11	224376048	-7183.32	1.797904310955119	1.902160541422583
2e11	424084653	-8611.89	1.800681441692738	1.902160557807120
3e11	615885700	-11077.61	1.802238425610275	1.902160567207455

4e11	802817718	–23093.64	1.803314522594797	1.902160635784981
5e11	986222314	–13877.92	1.804133287736837	1.902160595672241
6e11	1166916933	–18008.47	1.804792314084921	1.902160611601456
7e11	1345394380	–23486.68	1.805342641847864	1.902160627757985
8e11	1521998439	–22360.26	1.805814337650692	1.902160625580581
9e11	1696987738	–14624.66	1.806226588713636	1.902160608308483
1e12	1870585220	–25353.18	1.806592419175883	1.902160630437725
2e12	3552770943	46121.50	1.808931049664746	1.902160521956607
3e12	5173760785	34043.13	1.810246818931813	1.902160531376364
4e12	6756832076	–19173.40	1.811158095237466	1.902160561178057
5e12	8312493003	–17742.07	1.811852563407868	1.902160559627288
6e12	9846842484	4228.51	1.812412158420305	1.902160551021909
7e12	11363874338	–28648.11	1.812879924669885	1.902160561654484
8e12	12866256870	–50032.56	1.813281195346818	1.902160567365323
9e12	14356002120	–58481.45	1.813632156341301	1.902160569662763
1e13	15834664872	–66566.94	1.813943760684607	1.902160571080154
2e13	30198862775	–99750.43	1.815940298662853	1.902160579010763
3e13	44078684643	–172868.71	1.817066852499448	1.902160583828792
4e13	57657248284	–127115.12	1.817848459999912	1.902160581536889
5e13	71018282471	–72805.14	1.818444902933538	1.902160578986490
6e13	84209699420	8870.37	1.818926002994568	1.902160575928305
7e13	97262712867	–75675.28	1.819328479041238	1.902160578303203
8e13	110198743491	–68932.89	1.819673984268776	1.902160577983921
9e13	123033833767	–33794.18	1.819976357302832	1.902160577217768
1e14	135780321665	–56770.51	1.820244968130271	1.902160577783278

==

The slow growth of r_3(x) provides further evidence in favor of the Hardy-Littlewood approximation (2), which itself implies the truth of the conjecture of the infinitude of the twin primes.

Not shown are the counts of primes pi(x), maintained primarily for checking purposes; these are available at http://www.trnicely.net/pi/pix_0000.htm, at http://www.trnicely.net/pi/pix_0001.htm, and at this site. Note, however, that the generated data files include values for pi(x) for (1e9)(1e9)(1e14), which may well be one of the most extensive such listings in existence, and could be useful for other purposes.

As has been mentioned, the counts pi_2(x) and the corresponding partial sums B(x) so obtained agreed with all the values previously published by Brent (1975, 1976) to 1e11. Beyond that point, a number of the values of pi_2(x) and B(x) were checked (using an alternating exchange via electronic mail) against preliminary partial results obtained by M. Kutrib and J. Richstein (1996, 1997) in a similar calculation. All counts thus checked agreed, and all the partial sums agreed to at least 18 significant digits, with residual discrepancies believed to be caused by bugs in the gcc compiler used by the other party. Additional checking mechanisms were incorporated as previously explained.

One additional result of interest in this regard is the value of B(1e14) obtained from the ultra-precision arithmetic:

B(1e14) = 1.82024 49681 30270 52889 47178 38619 53382 83464 90782 17191 413 ,

with at least 41 decimals (and probably 47 decimals) believed to be correct.

Brun's constant and the error analysis

The extrapolated sum B*(1e14) is believed to produce the most accurate value known to date for Brun's constant,

(8)   B = B(+infinity) = 1.90216 05778 +/- 0.00000 00021

The stated error estimate corresponds to one standard deviation in the following sense. Based on the evidence presented below, it is believed that if the algorithm used to derive this error estimate sigma(x) is applied to a large number of randomly chosen values of x (for each of which B(x) is computed and then extrapolated to B*(x)), the true value B(+infinity) of Brun's constant will lie between B*(x) - sigma(x) and B*(x) + sigma(x) about 68 % of the time; between B*(x) - 2*sigma(x) and B*(x) + 2*sigma(x) about 95 % of the time; between B*(x) - 3*sigma(x) and B*(x) + 3*sigma(x) about 99 % of the time; etc., just as would be expected of the behavior of the standard deviation of a normally distributed variable.

The error estimate sigma(x) is computed as follows. Brent (1975) conjectures on theoretical grounds that the values of

(9)   P(x) = sqrt(x)*ln(x)*[B*(x) - B(+infinity)]

are asymptotically normally distributed with mean o(1) and standard deviation sqrt(8c_2). From the results of my computations, I evaluated P(x) at each of the (more than 105) tabulated data points, using B*(1e14) for B(+infinity). The resulting distribution was in reality only roughly normal, with mu = 0.01351, sigma_P = 0.6703, skewness a_3 = -0.6809, and kurtosis a_4 = 1.7859. The standard deviation sigma_P differed significantly from the value of 2.2981 predicted by Brent (1975). However, Brent (1975) based his ultimate error estimate not on the standard deviation of P(x), but on its absolute upper bound (over all tabulated points >= 1e4) of approximately 3.5.

Nonetheless, it seems preferable to base the error bound on the global behavior of P(x), rather than its behavior at one extreme point. Thus one might hope that the standard deviation (in some sense) sigma(x) of B*(x) - B(+infinity) could be deduced from (9) as

(10) $sigma(x) = sigma\_P(x)/(sqrt(x)*ln(x))$,

where sigma_P(x) is the standard deviation of the distribution of P(t) over all tabulated values of t < x, using B*(x) in place of B(+infinity). This would then provide the source of the final error estimate quoted in (8):

(11) $sigma(1e14) = 0.6703/(sqrt(1e14)*ln(1e14)) \sim 0.00000\ 00020\ 8$

For purposes of comparison, the error estimate generated using Brent's (1975) logic would be

(12) $E\_B = 3.495/(sqrt(1e14)*ln(1e14)) \sim 0.00000\ 00108\ 4$

since the maximum of abs(P(x)) would still occur at x = 860000 with a value nearly identical to Brent's.

Certainly, the use of equation (10) as thus interpreted merits a healthy dose of skepticism. As a practical test of its validity, I also applied it separately to each decade value of x in [1e4, 9e13], calculating sigma_P(x) for the distribution of P(t) over all tabulated values t < x (using B*(x) for B(+infinity)); and then from (10), sigma(x), the error estimate for B*(x) based on the data points < x. In other words, for each such decade value x, I computed an error estimate using the same algorithm which produced the error estimate quoted in (8). This error estimate was then compared with the best known estimate for the true error in B*(x), namely B*(x) - B*(1e14). Decade rather than linear intervals were chosen so that the sample would be logarithmically distributed and thus presumably less influenced by the fact that values near 1e14 would show artificially low error rates (due to the necessity of treating B*(1e14) as the best known value for B(+infinity)).

Partial results are shown in Table 2, which lists the values of x, sigma_P(x), sigma(x), and the error E(x) scaled to sigma(x):

(13) $E(x) = (B*(x) - B*(1e14))/sigma(x)$.

———

Table 2.
Success of sigma(x) as an error predictor.

x	sigma_P(x)	sigma(x)	E(x)
1e04	1.04287	1.13228e-03	1.26966
1e05	1.17395	3.22452e-04	0.00842
1e06	1.17952	8.53762e-05	-2.89571
1e07	1.46618	2.87656e-05	0.96245
1e08	1.30196	7.06790e-06	1.04135

1e09	1.29540	1.97673e-06	-0.17122
1e10	1.23791	5.37619e-07	-0.41209
8e10	1.00181	1.41084e-07	-1.25901
9e10	1.13094	1.49458e-07	-0.30094
1e11	1.11493	1.39200e-07	-0.26121
2e11	0.93614	8.04440e-08	-0.24832
3e11	0.85703	5.92089e-08	-0.17862
4e11	1.14191	6.75853e-08	0.85820
5e11	0.84107	4.41554e-08	0.40514
6e11	0.83010	3.95152e-08	0.85583
7e11	0.87371	3.82879e-08	1.30523
8e11	0.80947	3.30204e-08	1.44751
9e11	0.72422	2.77339e-08	1.10064
1e12	0.78024	2.82378e-08	1.86468
2e12	1.56969	3.91870e-08	-1.42462
3e12	1.15503	2.32114e-08	-1.99932
4e12	1.17522	2.02503e-08	-0.82000
5e12	1.03070	1.57638e-08	-1.15175
6e12	0.84056	1.16630e-08	-2.29455
7e12	0.96812	1.23716e-08	-1.30369
8e12	1.08938	1.29636e-08	-0.80363
9e12	1.11698	1.24824e-08	-0.65056
1e13	1.11802	1.18111e-08	-0.56753
2e13	1.08612	7.92982e-09	0.15479
3e13	1.15637	6.80338e-09	0.88860
4e13	0.89701	4.52841e-09	0.82890
5e13	0.82343	3.69178e-09	0.32592
6e13	0.95020	3.86663e-09	-0.47974
7e13	0.78574	2.94591e-09	0.17649
8e13	0.74248	2.59307e-09	0.07738
9e13	0.71818	2.35609e-09	-0.24002

==

Of the complete set of 90 values of x thus sampled, abs(E(x)) was less than sigma(x) in 62 cases (69 %); less than 2*sigma(x) in 86 cases (96 %); and less than 3*sigma(x) in all cases. Furthermore, varying the presumed ultimate value of B(+infinity) by as much as plus or minus sigma(1e14) from B*(1e14) produced no appreciable change in this distribution of errors; neither did changing the granularity of the data points from 1e9 to 1e0. Thus, sigma(x) indeed appears to be a reliable predictor of the error in B*(x) - B(+infinity), behaving in a manner similar to the standard deviation of a normally distributed variable.

Admittedly, this technique of error estimation is long on pragmatism and short on rigor; only a theoretical breakthrough or an enormous amount of additional computation will decide its ultimate fate. This is hardly unexpected when dealing with Brun's constant, the twin-primes conjecture, and the Hardy-Littlewood approximation, all of which have proved singularly resistant to analytical attack.

Acknowledgments

Literature Cited

- Bohman, Jan. 1973. Some computational results regarding the prime numbers below 2,000,000,000. Nordisk Tidskr. Informationsbehandling (BIT) 13:242-244; errata (1974) ibid. 14:127. MR 48 #217.
- Brent, Richard P. 1973. The first occurrence of large gaps between successive primes. Math. Comp. 27(124):959-963. MR 48 #8360.
- Brent, Richard P. 1975. Irregularities in the distribution of primes and twin primes. Math. Comp. 29(129):43-56, MR 51 #5522. Corrigendum (1976) ibid. 30(133):198, MR 53 #302. Addendum (1976) reviewed ibid. 30:379.
- Brun, Viggo. 1919. La série 1/5 + 1/7 + 1/11 + 1/13 + 1/17 + 1/19 + 1/29 + 1/31 + 1/41 + 1/43 + 1/59 + 1/61 + ..., où les dénominateurs sont 'nombres premieres jumeaux' est convergente ou finie. (French) Bulletin des sciences mathématiques 43:100-104, 124-128.
- Coe, Tim. 1995 (April). Inside the Pentium FDIV bug, Dr. Dobb's Journal 20(4:229):129-135.
- Fröberg, Carl-Erik. 1961. On the sum of inverses of primes and twin primes. Nordisk Tidskr. Informationsbehandling (BIT) 1:15-20.
- Hardy, G. H. and Littlewood, J. E. 1923. Some problems of 'Partitio Numerorum' III: On the expression of a number as a sum of primes, Acta Math. 44:1-70. See also *Collected papers of G. H. Hardy* (1966), Clarendon Press, Oxford I:561-630.
- Kutrib, Martin, and Jörg Richstein. 1996 (February). Primzahlen: Zwillinge aus dem Parallelrechner. (German) Spektrum der Wissenschaft, pp. 26-31.
- Lenstra, Arjen K. 1988-1991. The original version of this package, the ZBIGINT ultraprecision integer package developed, and graciously placed in the public domain, by Lenstra, Mark Riordan, and Marc Ringuette), is available (19 March 2001) at http://www.funet.fi/pub/crypt/cryptography/rpem/rpem/ (thanks to Charles Doty for this pointer).
- *PC Week*. 1995 (23 January). Intel, DEC, and Sun post strong fiscal quarters. 12(3):99.

- Richstein, Jörg. 1997. Unendliche Geschwisterliebe unter Zahlen oder Die Suche nach den Primzahlzwillingen. (German) Überblicke Mathematik 1996/1997, 135-143, Überbl. Math., Vieweg, Braunschweig, 1997. MR 99f:11117.
- Riesel, Hans. 1994. Prime numbers and computer methods for factorization. 2nd edition. Birkhäuser Boston. 464 pp. + xvi. ISBN 0-8176-3743-5. MR 95h:11142.
- Selmer, Ernst S. 1942. A special summation method in the theory of prime numbers and its application to "Brun's Sum." (Norwegian) Nordisk Mat. Tidskr. 24:74-81. MR 8,316g.
- Shanks, Daniel and Wrench, John W., Jr. 1974 (January). Brun's constant. Math. Comp. 28(125):293-299; corrigenda (1974) ibid. 28:1183. MR 50 #4510.
- Sharangpani, H. P. and Barton, M. L. 1994 (30 November). Statistical analysis of floating-point flaw in the Pentium processor (1994). Intel Corporation.
- Weintraub, Sol. 1973. Four tables concerning the distribution of primes. UMT 38. Math. Comp. 27:676-677.
- Wrench, John W., Jr. 1961. Evaluation of Artin's constant and the twin-prime constant. Math. Comp. 15:396-398. MR 23 #A1619.
- Young, Jeff and Potler, Aaron. 1989. First occurrence prime gaps. Math. Comp. 52(185):221-224. MR 89f:11019.

Unpublished Addendum

NOTE: This addendum is not part of the published journal article.

- At the present time (2009), the author believes the error analysis in the paper "Enumeration to 1.6e15 of the twin primes and Brun's constant" (1999) to be more accurate than the one presented in this paper. It obtains a considerably more conservative result, namely, that the error bound at any specific value of x is given by

    ```
    |B*(x) - B| < E_2(x) = 4.14083/(sqrt(x)*ln(x))
    ```

    See the referenced paper for details. By this formula, the result of the calculations up to 1e14 becomes

    ```
    B = 1.90216 05778 +/- 0.00000 00129
    ```

    Furthermore, although the cited paper represents this error bound as having a 99 % confidence level, the author is now (2009) of the opinion that this error bound is best characterized as representing only "at least one standard deviation," i.e., a confidence level of at least 68.27 %.
- The latest updated count of the twin primes, and the corresponding estimates for Brun's constant and a definitive error bound, are available at

- http://www.trnicely.net/counts.html.
- A more extensive version of Table 1, for 0 to 1e16, is available at http://www.trnicely.net/twins/t2_0000.htm. See also a corresponding table for the interval 1e16 to 2e16.
- Tables of the values of pi(x), determined by direct count as part of this project, are available for the interval 0 to 1e16 and for the interval 1e16 to 2e16.
- Complete counts and reciprocal sums of the prime constellations from Nicely's computations (1993-2009), including over two million data points from 0 to 2e16, are now available (these are very large files, over 60MB each even in zipped form).
- A similar project has been initiated by Pascal Sebah, with results available at http://numbers.computation.free.fr. Sebah has carried his computations to 1e16, with our results agreeing within rounding error.
- Another project similar to this one was initiated by Patrick Fry, Jeffrey Nesheiwat, and Boleslaw K. Szymanski, titled "Rensselaer's twin primes computing effort" (ca. 1997-2004). No final results of this work have been found; links to some preliminary and partial results have been collected at http://www.trnicely.net/twins/twins4.html#RTPCE.
- The reciprocal sums B(x) listed in Table 1 are believed to be correct, and correctly rounded, to all digits shown. The calculation of these quantities was carried out in long double precision (extended precision, 64-bit mantissa, 19 significant digits), using the Intel FPU hardware (numeric coprocessor). However, these calculations were also carried out to 53 decimal places in software, using the author's modification of the ZBIGINT ultraprecision integer package developed, and graciously placed in the public domain, by Arjen K. Lenstra, Mark Riordan, and Marc Ringuette (1988-1991). The original ZBIGINT package is available at http://www.funet.fi/pub/crypt/cryptography/rpem/rpem/ (thanks to Charles Doty for this pointer). The predecessor of this package was Lenstra's LIP (long integer package). Its descendant is NTL (Number Theory Library), maintained by Victor Shoup. However, these packages have to a large extent been rendered obsolete by GMP (GNU MP), the GNU multiple precision library, and MPFR, a C library for multiple-precision floating-point computations with correct rounding, reliable precision control, and compatibility with the ANSI/IEEE 754-1985 standard.

- [Complete counts and reciprocal sums of the prime constellations from Nicely's computations](#) (tables)
- [Home Page](#)