The Quest for Pi

D. H. BAILEY, J. M. BORWEIN, P. B. BORWEIN, AND S. PLOUFFE

This article gives a brief history of the analysis and computation of the mathematical constant $\pi = 3.14159\ldots$, including a number of formulas that have been used to compute π through the ages. Some exciting recent developments are then discussed in some detail, including the recent computation of π to over six billion decimal digits using high-order convergent algorithms, and a newly discovered scheme that permits arbitrary individual hexadecimal digits of π to be computed.

For further details of the history of π up to about 1970, the reader is referred to Petr Beckmann's readable and entertaining book [3]. A listing of milestones in the history of the computation of π is given in Tables 1 and 2, which we believe to be more complete than other readily accessible sources.

The Ancients

In one of the earliest accounts (about 2000 B.C.) of π, the Babylonians used the approximation $3\frac{1}{8} = 3.125$. At this same time or earlier, according to an account in an ancient Egyptian document, Egyptians were assuming that a circle with diameter nine has the same area as a square of side eight, which implies $\pi = 256/81 = 3.1604\ldots$. Others of antiquity were content to use the simple approximation 3, as evidenced by the following passage from the Old Testament:

> Also, he made a molten sea of ten cubits from brim to brim, round in compass, and five cubits the height thereof; and a line of thirty cubits did compass it round about. (I Kings 7:23; see also 2 Chron. 4:2)

The first rigorous mathematical calculation of the value of π was due to Archimedes of Syracuse (~250 B.C.), who used a geometrical scheme based on inscribed and circumscribed polygons to obtain the bounds $3\frac{10}{71} < \pi < 3\frac{1}{7}$, or in other words, $3.1408\ldots < \pi < 3.1428\ldots$ [11]. No one was able to improve on Archimedes's method for many centuries, although a number of persons used this general method to obtain more accurate approximations. For example, the astronomer Ptolemy, who lived in Alexandria in A.D. 150, used the value $3\frac{17}{120} = 3.141666\ldots$, and the fifth-century Chinese mathematician Tsu Chung-Chih used a variation of Archimedes's method to compute π correct to seven digits, a level not attained in Europe until the 1500s.

The Age of Newton

As in other fields of science and mathematics, progress in the quest for π in medieval times occurred mainly in the Islamic world. Al-Kashi of Samarkand computed π to 14 places in about 1430.

In the 1600s, with the discovery of calculus by Newton

TABLE 1. History of π Calculations (Pre-20th-Century)

Babylonians	2000? B.C.E.	1	3.125 ($3\frac{1}{8}$)
Egyptians	2000? B.C.E.	0	3.16045 [$4(\frac{8}{9})^2$]
China	1200? B.C.E.	0	3
Bible (1 Kings 7:23)	550? B.C.E.	0	3
Archimedes	250? B.C.E.	3	3.1418 (ave.)
Hon Han Shu	A.D. 130	0	3.1622 (= $\sqrt{10}$?)
Ptolemy	150	3	3.14166
Chung Hing	250?	0	3.16227 ($\sqrt{10}$)
Wang Fau	250?	0	3.15555 ($\frac{142}{45}$)
Liu Hui	263	5	3.14159
Siddhanta	380	4	3.1416
Tsu Ch'ung Chi	480?	7	3.1415926
Aryabhata	499	4	3.14156
Brahmagupta	640?	0	3.162277 (= $\sqrt{10}$)
Al-Khowarizmi	800	4	3.1416
Fibonacci	1220	3	3.141818
Al-Kashi	1429	14	
Otho	1573	6	3.1415929
Viète	1593	9	3.1415926536 (ave.)
Romanus	1593	15	
Van Ceulen	1596	20	
Van Ceulen	1615	35	
Newton	1665	16	
Sharp	1699	71	
Seki	1700?	10	
Kamata	1730?	25	
Machin	1706	100	
De Lagny	1719	127	(112 correct)
Takebe	1723	41	
Matsunaga	1739	50	
Vega	1794	140	
Rutherford	1824	208	(152 correct)
Strassnitzky and Dase	1844	200	
Clausen	1847	248	
Lehmann	1853	261	
Rutherford	1853	440	
Shanks	1874	707	(527 correct)

TABLE 2. History of π Calculations (20th Century)

Ferguson	1946	620
Ferguson	Jan. 1947	710
Ferguson and Wrench	Sep. 1947	808
Smith and Wrench	1949	1,120
Reitwiesner, *et al.* (ENIAC)	1949	2,037
Nicholson and Jeenel	1954	3,092
Felton	1957	7,480
Genuys	Jan. 1958	10,000
Felton	May 1958	10,021
Guilloud	1959	16,167
Shanks and Wrench	1961	100,265
Guilloud and Filliatre	1966	250,000
Guilloud and Dichampt	1967	500,000
Guilloud and Bouyer	1973	1,001,250
Miyoshi and Kanada	1981	2,000,036
Guilloud	1982	2,000,050
Tamura	1982	2,097,144
Tamura and Kanada	1982	4,194,288
Tamura and Kanada	1982	8,388,576
Kanada, Yoshino, and Tamura	1982	16,777,206
Ushiro and Kanada	Oct. 1983	10,013,395
Gosper	1985	17,526,200
Bailey	Jan. 1986	29,360,111
Kanada and Tamura	Sep. 1986	33,554,414
Kanada and Tamura	Oct. 1986	67,108,839
Kanada, Tamura, Kubo, *et al.*	Jan. 1987	134,217,700
Kanada and Tamura	Jan. 1988	201,326,551
Chudnovskys	May 1989	480,000,000
Chudnovskys	June 1989	525,229,270
Kanada and Tamura	July 1989	536,870,898
Kanada and Tamura	Nov. 1989	1,073,741,799
Chudnovskys	Aug. 1989	1,011,196,691
Chudnovskys	Aug. 1991	2,260,000,000
Chudnovskys	May 1994	4,044,000,000
Takahashi and Kanada	June 1995	3,221,225,466
Kanada	Aug. 1995	4,294,967,286
Kanada	Oct. 1995	6,442,450,938

and Leibniz, a number of substantially new formulas for π were discovered. One of them can be easily derived by recalling that

$$\tan^{-1} x = \int_0^x \frac{dt}{1+t^2} = \int_0^x (1 - t^2 + t^4 - t^6 + \cdots)\, dt$$
$$= x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \cdots.$$

Substituting $x = 1$ gives the well-known Gregory–Leibniz formula

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots.$$

Regrettably, this series converges so slowly that hundreds of terms would be required to compute the numerical value of π to even two digits accuracy. However, by employing the trigonometric identity

$$\frac{\pi}{4} = \tan^{-1}\left(\frac{1}{2}\right) + \tan^{-1}\left(\frac{1}{3}\right)$$

(which follows from the addition formula for the tangent function), one obtains

$$\frac{\pi}{4} = \left(\frac{1}{2} - \frac{1}{3\cdot 2^3} + \frac{1}{5\cdot 2^5} - \frac{1}{7\cdot 2^7} + \cdots\right)$$
$$+ \left(\frac{1}{3} - \frac{1}{3\cdot 3^3} + \frac{1}{5\cdot 3^5} - \frac{1}{7\cdot 3^7} + \cdots\right),$$

which converges much more rapidly. An even faster formula, due to Machin, can be obtained by employing the identity

$$\frac{\pi}{4} = 4\tan^{-1}\left(\frac{1}{5}\right) - \tan^{-1}\left(\frac{1}{239}\right)$$

in a similar way. Shanks used this scheme to compute π to 707 decimal digits accuracy in 1873. Alas, it was later found that this computation was in error after the 527th decimal place.

Newton discovered a similar series for the arcsine function:

$$\sin^{-1}x = x + \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \cdots.$$

π can be computed from this formula by noting that $\pi/6 = \sin^{-1}(1/2)$. An even faster formula of this type is

$$\pi = \frac{3\sqrt{3}}{4} + 24\left(\frac{1}{3 \cdot 2^3} - \frac{1}{5 \cdot 2^5} - \frac{1}{7 \cdot 2^7} - \frac{1}{9 \cdot 2^9} - \cdots\right).$$

Newton himself used this particular formula to compute π. He published only 15 digits, but later sheepishly admitted, "I am ashamed to tell you how many figures I carried these computations, having no other business at the time."

In the 1700s, the mathematician Euler, arguably the most prolific mathematician in history, discovered a number of new formulas for π. Among these are

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \cdots,$$
$$\frac{\pi^4}{90} = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \frac{1}{5^4} + \cdots.$$

A related, more rapidly convergent series is

$$\frac{\pi^2}{6} = 3\sum_{m=1}^{\infty} \frac{1}{m^2\binom{2m}{m}}.$$

These formulas, despite their important theoretical implications, aren't very efficient for computing π.

One motivation for computations of π during this time was to see if the decimal expansion of π repeats, thus disclosing that π is the ratio of two integers (although hardly anyone in modern times seriously believed this). The question was settled in the late 1700s, when Lambert and Legendre proved that π is irrational. Some still wondered whether π might be the root of some algebraic equation with integer coefficients (although, as before, few really believed that it was). This question was finally settled in 1882 when Lindemann proved that π is transcendental. Lindemann's proof also settled once and for all, in the negative, the ancient Greek question of whether the circle could be squared with rule and compass. This is because constructible numbers are necessarily algebraic.

In the annals of π, the march of the nineteenth-century progress sometimes faltered. Three years prior to the turn of the century, one Edwin J. Goodman, M.D. introduced into the Indiana House of Representatives a "new Mathematical truth" to enrich the state, which would profit from the royalties ensuing from this discovery. Section two of his bill included the passage

> disclosing the fourth important fact that the ratio of the diameter and circumference is as five-fourths to four;

Thus, one of Goodman's new mathematical "truths" is that $\pi = \frac{16}{5} = 3.2$. The Indiana House passed the bill unanimously on Feb. 5, 1897. It then passed a Senate committee and would have been enacted into law had it not been for the last-minute intervention of Prof. C. A. Waldo of Purdue University, who happened to hear some of the deliberation while on other business.

The Twentieth Century

With the development of computer technology in the 1950s, π was computed to thousands and then millions of digits, in both decimal and binary bases (see, for example, [17]). These computations were facilitated by the discovery of some advanced algorithms for performing the required high-precision arithmetic operations on a computer. For example, in 1965, it was found that the newly discovered fast Fourier transform (FFT) could be used to perform high-precision multiplications much more rapidly than conventional schemes. These methods dramatically lowered the computer time required for computing π and other mathematical constants to high precision. See [1], [7], and [8].

In spite of these advances, until the 1970s all computer evaluations for π still employed classical formulas, usually a variation of Machin's formula. Some new infinite series formulas were discovered by the Indian mathematician Ramanujan around 1910, but these were not well known until quite recently when his writings were widely published. One of these is the remarkable formula

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801}\sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26{,}390k)}{(k!)^4 396^{4k}}.$$

Each term of this series produces an additional eight correct digits in the result. Gosper used this formula to compute 17 million digits of π in 1985.

Although Ramanujan's series is considerably more efficient than the classical formulas, it shares with them the property that the number of terms one must compute increases linearly with the number of digits desired in the result. In other words, if one wishes to compute π to twice as many digits, then one must evaluate twice as many terms of the series.

In 1976, Eugene Salamin [16] and Richard Brent [8] independently discovered a new algorithm for π, which is based on the arithmetic–geometric mean and some ideas originally due to Gauss in the 1800s (although, for some reason, Gauss never saw the connection to computing π). This algorithm produces approximations that converge to π much more rapidly than any classical formula. The Salamin–Brent algorithm may be stated as follows. Set $a_0 = 1$, $b_0 = 1/\sqrt{2}$, and $s_0 = 1/2$. For $k = 1, 2, 3, \ldots$ compute

$$a_k = \frac{a_{k-1} + b_{k-1}}{2},$$
$$b_k = \sqrt{a_{k-1}b_{k-1}},$$
$$c_k = a_k^2 - b_k^2,$$
$$s_k = s_{k-1} - 2^k c_k,$$
$$p_k = \frac{2a_k^2}{s_k}.$$

Then p_k converges *quadratically* to π. This means that each iteration of this algorithm approximately *doubles* the number of correct digits. To be specific, successive iterations produce 1, 4, 9, 20, 42, 85, 173, 347, and 697 correct digits of π. Twenty-five iterations are sufficient to compute π to over 45 million decimal digit accuracy. However, each of these iterations must be performed using a level of numeric precision that is at least as high as that desired for the final result.

The Salamin–Brent algorithm requires the extraction of square roots to high precision, operations not required, for example, in Machin's formula. High-precision square roots can be efficiently computed by means of a Newton iteration scheme that employs only multiplications, plus some other operations of minor cost, using a level of numeric precision that doubles with each iteration. The total cost of computing a square root in this manner is only about three times the cost of performing a single full-precision multiplication. Thus, algorithms such as the Salamin–Brent scheme can be implemented very rapidly on a computer.

Beginning in 1985, two of the present authors (Jonathan and Peter Borwein) discovered some additional algorithms of this type [5–7]. One is as follows. Set $a_0 = 1/3$ and $s_0 = (\sqrt{3} - 1)/2$. Iterate

$$r_{k+1} = \frac{3}{1 + 2(1 - s_k^3)^{1/3}},$$
$$s_{k+1} = \frac{r_{k+1} - 1}{2},$$
$$a_{k+1} = r_{k+1}^2 a_k - 3^k(r_{k+1}^2 - 1).$$

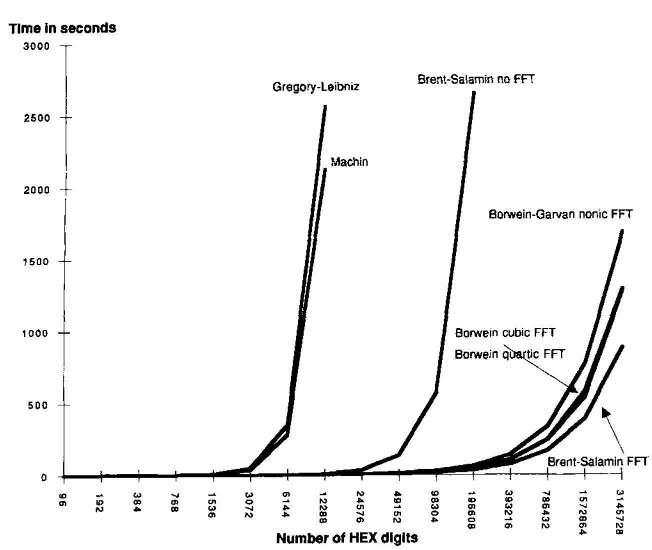
Then $1/a_k$ converges *cubically* to π—each iteration approximately triples the number of correct digits.

A quartic algorithm is as follows: Set $a_0 = 6 - 4\sqrt{2}$ and $y_0 = \sqrt{2} - 1$. Iterate

$$y_{k+1} = \frac{1 - (1 - y_k^4)^{1/4}}{1 + (1 - y_k^4)^{1/4}},$$
$$a_{k+1} = a_k(1 + y_{k+1})^4 - 2^{2k+3} y_{k+1}(1 + y_{k+1} + y_{k+1}^2).$$

Then $1/a_k$ converges *quartically* to π. This particular algorithm, together with the Salamin–Brent scheme, has

FIGURE 1



been employed by Yasumasa Kanada of the University of Tokyo in several computations of π over the past 10 years or so. In the latest of these computations, Kanada computed over 6.4 billion decimal digits on a Hitachi supercomputer. This is presently the world's record in this arena.

More recently, it has been further shown that there are algorithms that generate mth-order convergent approximations to π for any m. An example of a nonic (ninth-order) algorithm is the following: Set $a_0 = 1/3$, $r_0 = (\sqrt{3} - 1)/2$, and $s_0 = (1 - r_0^3)^{1/3}$. Iterate

$$t = 1 + 2r_k,$$
$$u = [9r_k(1 + r_k + r_k^2)]^{1/3},$$
$$v = t^2 + tu + u^2,$$
$$m = \frac{27(1 + s_k + s_k^2)}{v},$$
$$a_{k+1} = ma_k + 3^{2k-1}(1 - m),$$
$$s_{k+1} = \frac{(1 - r_k)^3}{(t + 2u)v},$$
$$r_{k+1} = (1 - s_k^3)^{1/3}.$$

Then $1/a_k$ converges *nonically* to π. It should be noted, however, that these higher-order algorithms do not appear to be faster as computational schemes than, say, the Salamin–Brent or the Borwein quartic algorithms. Although fewer iterations are required to achieve a given level of precision in the higher-order schemes, each iteration is more expensive.

A comparison of actual computer run times for various π algorithms is shown in Figure 1. These run times are for computing π in binary to various precision levels on an IBM RS6000/590 workstation. The abscissa of this plot is in hexadecimal digits—multiply these numbers by 4 to obtain equivalent binary digits, or by $\log_{10}(16) = 1.20412\ldots$ to obtain equivalent decimal digits. Other implementations on other systems may give somewhat different results—for example, in Kanada's recent computation of π to over six billion digits, the quartic algorithm ran somewhat faster than the Salamin–Brent algorithm (116 hours versus 131 hours). But the overall picture from such comparisons is unmistakable: the modern schemes run many times faster than the classical schemes, especially when implemented using FFT-based arithmetic.

David and Gregory Chudnovsky of Columbia University have also done some very high-precision computations of π in recent years, alternating with Kanada for the world's record. Their most recent computation (1994) produced over four billion digits of π [9]. They did not employ a high-order convergent algorithm, such as the Salamin–Brent or Borwein algorithms, but instead utilized the following infinite series (which is in the spirit of Ramanujan's series above):

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k\,(6k)!(13{,}591{,}409 + 545{,}140{,}134k)}{(3k)!\,(k!)^3\,640{,}320^{3k+3/2}}$$

Each term of this series produces an additional 14 correct digits. The Chudnovskys implemented this formula with a very clever scheme that enabled them to utilize the results of a certain level of precision to extend the calculation to even higher precision. Their program was run on a home-brewed supercomputer that they have assembled using private funds. An interesting personal glimpse of the Chudnovsky brothers is given in [14].

Computing Individual Digits of π

At several junctures in the history of π, it was widely believed that virtually everything of interest with regard to this constant had been discovered and, in particular, that no fundamentally new formulas for π lay undiscovered. This sentiment was even suggested in the closing chapters of Beckmann's 1971 book on the history of π [3], p. 172. Ironically, the Salamin–Brent algorithm was discovered only 5 years later.

A more recent reminder that we have not come to the end of humanity's quest for knowledge about π came with the discovery of the Rabinowitz–Wagon "spigot" algorithm for π in 1990 [15]. In this scheme, successive digits of π (in any desired base) can be computed with a relatively simple recursive algorithm based on the previously generated digits. Multiple-precision computation software is not required; therefore, this scheme can be easily implemented on a personal computer.

Note, however, that this algorithm, like all of the other schemes mentioned above, still has the property that in order to compute the dth digit of π, one must first (or simultaneously) compute each of the preceding digits. In other words, there is no "shortcut" to computing the dth digit with these formulas. Indeed, it has been widely assumed in the field (although never proven) that the computational complexity of computing the dth digit is not significantly less than that of computing all of the digits up to and including the dth digit. This may still be true, although it is probably very hard to prove. Another common feature of the previously known π algorithms is that they all appear to require substantial amounts of computer memory, amounts that typically grow linearly with the number of digits generated.

Thus, it was with no small surprise that a novel scheme was recently discovered for computing individual hexadecimal digits of π [2]. In particular, this algorithm (1) produces the dth hexadecimal (base 16) digit of π directly, without the need of computing any previous digits, (2) is quite simple to implement on a computer, (3) does not require multiple-precision arithmetic software, (4) requires very little memory, and (5) has a computational cost that grows only slightly faster than the index d. For example, the one millionth hexadecimal digit of π can be computed in only a minute or two on a current RISC workstation or high-end personal computer. This algorithm is not fundamentally faster than other known schemes for computing all digits up to some position d, but its elegance and simplicity are, nonetheless, of considerable interest.

This scheme is based on the following remarkable new formula for π:

$$\pi = \sum_{i=0}^{\infty} \frac{1}{16^i}\left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6}\right).$$

The proof of this formula is not very difficult. First, note that for any $k < 8$,

$$\int_0^{1/\sqrt{2}} \frac{x^{k-1}}{1-x^8}\,dx = \int_0^{1/\sqrt{2}} \sum_{i=0}^{\infty} x^{k-1+8i}\,dx = \frac{1}{2^{k/2}} \sum_{i=0}^{\infty} \frac{1}{16^i(8i+k)}.$$

Thus, we can write

$$\sum_{i=0}^{\infty} \frac{1}{16^i}\left(\frac{4}{8i+1} - \frac{2}{8i+4} - \frac{1}{8i+5} - \frac{1}{8i+6}\right) = \int_0^{1/\sqrt{2}} \frac{4\sqrt{2} - 8x^3 - 4\sqrt{2}x^4 - 8x^5}{1-x^8}\,dx,$$

which on substituting $y := \sqrt{2}x$ becomes

$$\int_0^1 \frac{16y-16}{y^4 - 2y^3 + 4y - 4}\,dy = \int_0^1 \frac{4y}{y^2-2}\,dy - \int_0^1 \frac{4y-8}{y^2-2y+2}\,dy = \pi,$$

reflecting a partial fraction decomposition of the integral on the left-hand side.

However, this derivation is dishonest, in the sense that the actual route of discovery was much different. This formula was actually discovered not by formal reasoning, but instead by numerical searches on a computer using the "PSLQ" integer-relation-finding algorithm [10]. Only afterward was a proof found.

A similar formula for π^2 (which also was first discovered using the PSLQ algorithm) is as follows:

$$\pi^2 = \sum_{i=0}^{\infty} \frac{1}{16^i}\left(\frac{16}{(8i+1)^2} - \frac{16}{(8i+2)^2} - \frac{8}{(8i+3)^2} - \frac{16}{(8i+4)^2} - \frac{4}{(8i+5)^2} - \frac{4}{(8i+6)^2} + \frac{2}{(8i+7)^2}\right).$$

Formulas of this type for a few other mathematical constants are given in [2].

Computing individual hexadecimal digits of π using the above formula crucially relies on what is known as the binary algorithm for exponentiation, wherein one evaluates x^n by successive squaring and multiplication. This reduces the number of multiplications required to less than 2 $\log_2(n)$. According to Knuth, this technique dates back at least to 200 B.C. [13]. In our application, we need to obtain the exponentiation result modulo a positive integer c. This can be efficiently done with the following variant of the binary exponentiation algorithm, wherein the result of each multiplication is reduced modulo c:

To compute $r = b^n \bmod c$, first set t to be the largest power of $2 \leq n$, and set $r = 1$. Then

A: if $n \geq t$ then $r \leftarrow br \bmod c$; $n \leftarrow n - t$; endif
$t \leftarrow t/2$
if $t \geq 1$ then $r \leftarrow r^2 \bmod c$; go to A; endif

Upon exit from this algorithm, r has the desired value. Here "mod" is used in the binary operator sense, namely as the binary function defined by $x \bmod y := x - [x/y]y$. Note that the above algorithm is entirely performed with positive integers that do not exceed c^2 in size. As an example, when computing 3^{49} mod 400 by this scheme, the variable r assumes the values 1, 9, 27, 329, 241, 81, 161, 83. Indeed 3^{49} = 239299329230617529590083, so that 83 is the correct result.

Consider now the first of the four sums in the formula above for π:

$$S_1 = \sum_{k=0}^{\infty} \frac{1}{16^k(8k+1)}.$$

First observe that the hexadecimal digits of S_1 beginning at position $d + 1$ can be obtained from the fractional part of $16^d S_1$. Then we can write

$$\begin{aligned}\text{frac}(16^d S_1) &= \sum_{k=0}^{\infty} \frac{16^{d-k}}{8k+1} \bmod 1 \\ &= \sum_{k=0}^{d} \frac{16^{d-k} \bmod 8k+1}{8k+1} \bmod 1 \\ &+ \sum_{k=d+1}^{\infty} \frac{16^{d-k}}{8k+1} \bmod 1.\end{aligned}$$

For each term of the first summation, the binary exponentiation scheme can be used to rapidly evaluate the numerator. In a computer implementation, this can be done using either integer or 64-bit floating-point arithmetic. Then floating-point arithmetic can be used to perform the division and add the quotient to the sum mod 1. The second summation, where the exponent of 16 is negative, may be evaluated as written using floating-point arithmetic. It is only necessary to compute a few terms of this second summation, just enough to ensure that the remaining terms sum to less than the "epsilon" of the floating-point arithmetic being used. The final result, a fraction between 0 and 1, is then converted to base 16, yielding the $(d + 1)$th hexadecimal digit, plus several additional digits. Full details of this scheme, including some numerical considerations, as well as analogous formulas for a number of other basic mathematical constants, can be found in [2]. Sample implementations of this scheme in both Fortran and C are available from the web site *http://www.cecm.sfu.ca/personal/pborwein/*.

As the reader can see, there is nothing very sophisticated about either this new formula for π, its proof, or the scheme just described to compute hexadecimal digits of π using it. In fact, this same scheme can be used to compute binary (or hexadecimal) digits of log(2) based on the formula

$$\log(2) = \sum_{k=1}^{\infty} \frac{1}{k2^k},$$

which has been known for centuries. Thus, it is astonishing that these methods have lain undiscovered all this time. Why shouldn't Euler, for example, have discovered them? The only advantage that today's researchers have in this regard is advanced computer technology. Table 3 gives some hexadecimal digits of π computed using the above scheme.

One question that immediately arises is whether or not there is a formula of this type and an associated computational scheme to compute individual *decimal* digits of π. Alas, no decimal scheme for π is known at this time, although there is for certain constants such as log(9/10)—see [2]. On the other hand, there is not yet any proof that a decimal scheme for π cannot exist. This question is currently being actively pursued. Based on some numerical searches using the PSLQ algorithm, it appears that there are no simple formulas for π of the above form with 10 in the place of 16. This, of course, does not rule out the possibility of completely different formulas that nonetheless permit rapid computation of individual decimal digits of π.

TABLE 3. Hexadecimal Digits of π

Position	Hex digits beginning at this position
10^6	26C65E52CB4593
10^7	17AF5863EFED8D
10^8	ECB840E21926EC
10^9	85895585A0428B
10^{10}	921C73C6838FB2

Fabrice Bellard tells us that he recently completed the computation of the 100 billion'th hexadecimal digit by this method, this gives:

9C381872D27596F81D0E. . .

Why?

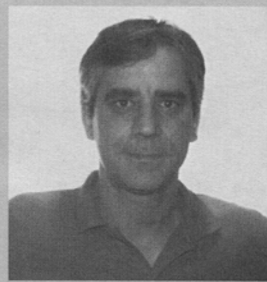
A value of π to 40 digits would be more than enough to compute the circumference of the Milky Way galaxy to an error less than the size of a proton. There are certain scientific calculations that require intermediate calculations to be performed to significantly higher precision than required for the final results, but it is doubtful that anyone will ever need more than a few hundred digits of π for such purposes. Values of π to a few thousand digits are sometimes employed in explorations of mathematical questions using a computer, but we are not aware of any significant applications beyond this level.

One motivation for computing digits of π is that these calculations are excellent tests of the integrity of computer hardware and software. This is because if even a single error occurs during a computation, almost certainly the final result will be in error. On the other hand, if two independent computations of digits of π agree, then most likely both computers performed billions or even trillions of operations flawlessly. For example, in 1986, a π-calculating program detected some obscure hardware problems in one of the original Cray-2 supercomputers [1].

The challenge of computing π has also stimulated research into advanced computational techniques. For example, some new techniques for efficiently computing linear convolutions and fast Fourier transforms, which have applications in many areas of science and engineering, had their origins in efforts to accelerate computations of π.

Beyond immediate practicality, decimal and binary expansions of π have long been of interest to mathematicians, who have still not been able to resolve the question of whether the expansion of π is normal [18]. In particular, it is widely suspected that the decimal expansions of π, e, $\sqrt{2}$, $\sqrt{10}$, and many other mathematical constants all have the property that the limiting frequency of any digit is one-tenth, and the limiting frequency of any n-long string

AUTHORS



DAVID H. BAILEY
NASA Ames Research Center
Moffett Field, CA 94035-1000, USA
dbailey@nas.nasa.gov

SIMON M. PLOUFFE
Centre for Experimental and Constructive Mathematics
Simon Fraser University
Burnaby, BC V5A 1S6, Canada
plouffe@cecm.sfu.ca

PETER B. BORWEIN
Centre for Experimental and Constructive Mathematics
Simon Fraser University
Burnaby, BC V5A 1S6, Canada
pborwein@cecm.sfu.ca

JONATHAN M. BORWEIN
Centre for Experimental and Constructive Mathematics
Simon Fraser University
Burnaby, BC V5A 1S6, Canada
jborwein@cecm.sfu.ca

David H. Bailey received his Ph.D. from Stanford in mathematics. Currently he is with the Numerical Aerodynamic Simulation Program at NASA Ames Research Center. His research has included studies in numerical analysis, computational number theory, and supercomputer performance. He has received the Sid Fernbach Award from the IEEE Computer Society, and shared the Chauvenet and Merten Hasse Prizes from the Mathematical Association of America, with Peter and Jonathan Borwein.

Simon Plouffe (http://www.cecm.sfu.ca/~plouffe) is currently a Research Associate at the CECM. He recently found with D.H. Bailey and Peter Borwein an algorithm for the computation of the n'th binary digit of Pi. He is a co-author with Neil J.A. Sloane of the *Encyclopedia of Integer Sequences* and is now in charge of the Inverse Symbolic Calculator project at http://www.cecm.sfu.ca/projects/ISC.

Jonathan Borwein (http://www.cecm.sfu.ca/~jborwein) and Peter Borwein (http://www.cecm.sfu.ca/~pborwein) have provided fuller biographic information in a recent article on Experimental Mathematics in *The Intelligencer.* They direct the Centre for Experimental and Constructive Mathematics at which Simon Plouffe works.

of decimal digits is 10^{-n} (and similarly for binary expansions). Such a guaranteed property could, for instance, be the basis of a reliable pseudo-random-number generator for scientific calculations. Unfortunately, this assertion has not been proven in even one instance. Thus, there is a continuing interest in performing statistical analyses on the expansions of these numbers to see if there is any irregularity that would make them look unlike random sequences. So far, such studies of high-precision values of π have not disclosed any irregularities. Along this line, new formulas and schemes for computing digits of π are of interest because they may suggest new approaches to the normality question.

Finally, there is a more fundamental motivation for computing π, the challenge, like that of a lofty mountain or a major sporting event: "it is there." π is easily the most famous of the basic constants of mathematics. Every technical civilization has to master π, and we wonder if it may be equally inevitable that someone feels the challenge to raise the precision of its computation.

The constant π has repeatedly surprised humanity with new and unanticipated results. If anything, the discoveries of this century have been even more startling, with respect to the previous state of knowledge, than those of past centuries. We guess from this that even more surprises lurk in the depths of undiscovered knowledge regarding this famous constant.

ACKNOWLEDGMENT

The authors wish to acknowledge helpful information from Yasumasa Kanada of the University of Tokyo.

REFERENCES

1. D. H. Bailey, The computation of pi to 29,360,000 decimal digits using Borweins' quartically convergent algorithm, *Mathematics of Computation* 42 (1988), 283–296.
2. D. H. Bailey, P. B. Borwein, and S. Plouffe, On the rapid computation of various polylogarithmic constants, (to appear in Mathematics of Computation). Available from *http://www.cecm.sfu/personal/pborwein/.*
3. P. Beckmann, *A History of Pi*, New York: St. Martin's Press (1971).
4. L. Berggren, J. M. Borwein, and P. B. Borwein, *A Sourcebook on Pi*, New York: Springer-Verlag (to appear).
5. J. M. Borwein and P. B. Borwein, *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*, New York: Wiley (1987).
6. J. M. Borwein and P. B. Borwein, Ramanujan and pi, *Scientific American* (February 1987), 112–117.
7. J. M. Borwein, P. B. Borwein, and D. H. Bailey, Ramanujan, modular equations, and approximations to pi, or how to compute one billion digits of pi, *American Mathematical Monthly* 96 (1989), 201–219. Also available from the URL *http://www.cecm.sfu.ca/personal/pborwein/.*
8. R. P. Brent, Fast multiple-precision evaluation of elementary functions, *Journal of the ACM* 23 (1976), 242–251.
9. D. Chudnovsky and C. Chudnovsky, personal communication (1995).
10. H. R. P. Ferguson and D. H. Bailey, Analysis of PSLQ, an integer relation algorithm, unpublished, 1996.