Some Background on Kanada's Recent Pi Calculation

David H. Bailey

16 May 2003

History

I will first give some historical background, condensed from [1]. Several of the commonly used algorithms for calculating π have their roots in classical arctangent-based formulas, such as

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

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

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \cdots$$

This particular series is useless for computing π — it converges so slowly that hundreds of terms would be required to compute even two correct digits. However, by employing the trigonometric identity

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

you can obtain

$$\begin{aligned}\pi/4 &= \frac{1}{2} - \frac{1}{3 \cdot 2^3} + \frac{1}{5 \cdot 2^5} - \frac{1}{7 \cdot 2^7} + \cdots \\ &\quad + \frac{1}{3} - \frac{1}{3 \cdot 3^3} + \frac{1}{5 \cdot 3^5} - \frac{1}{7 \cdot 3^7} + \cdots\end{aligned}$$

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

$$\pi/4 = 4 \tan^{-1}(1/5) - \tan^{-1}(1/239)$$

in a similar way. This formula was used in numerous computations of π, culminating with Shanks' computation of π to 707 decimal digits accuracy in 1873 (although it was later found that this result was in error after the 527-th decimal place).

With the development of computer technology in the 1950s, π was computed to thousands and then millions of digits. These computations were facilitated by the discovery that high-precision multiplication could be performed rapidly using fast Fourier transform (FFT)s. In spite of these advances, until the 1970s all computer evaluations of π still employed classical formulas, usually one of the Machin-type formulas. Some new infinite series formulas were discovered by Ramanujan around 1910, but these were not well known until quite recently when his writings were widely published. One of these is the formula

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

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. At about the same time, David and Gregory Chudnovsky found the following variation of Ramanujan's formula:

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k (6k)!(13591409 + 545140134k)}{(3k)!\,(k!)^3\, 640320^{3k+3/2}}$$

Each term of this series produces an additional 14 correct digits. The Chudnovskys implemented this formula using a clever scheme that enabled them to utilize the results of an initial level of precision to extend the calculation to even higher precision. They used this method in several large calculations of π, culminating with a computation to over four billion decimal digits in 1994.

While the Ramanujan and Chudnovsky series are considerably more efficient than the classical formulas, they share 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 you want to compute π to twice as many digits, you have to evaluate twice as many terms of the series.

In 1976, Eugene Salamin and Richard Brent independently discovered an algorithm for π based on the arithmetic-geometric mean (AGM) and some ideas originally due to Gauss in the 1800s (although for some reason Gauss never saw the connection to computing π). The Salamin–Brent algorithm may be stated as follows. Set $a_0 = 1, b_0 = 1/\sqrt{2}$ and $s_0 = 1/2$. Calculate

$$\begin{aligned} a_k &= \frac{a_{k-1} + b_{k-1}}{2} \\ b_k &= \sqrt{a_{k-1} b_{k-1}} \\ c_k &= a_k^2 - b_k^2 \\ s_k &= s_{k-1} - 2^k c_k \\ p_k &= \frac{2a_k^2}{s_k} \end{aligned}$$

Then p_k converges *quadratically* to π: each iteration of this algorithm approximately doubles the number of correct digits — successive iterations produce 1, 4, 9, 20, 42, 85, 173, 347 and 697 correct decimal 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.

Beginning in 1985, Canadian mathematicians Jonathan Borwein and Peter Borwein discovered some additional algorithms of this type. One of these is as follows: Set $a_0 = 6 - 4\sqrt{2}$ and $y_0 = \sqrt{2} - 1$. Iterate

$$\begin{aligned} y_{k+1} &= \frac{1 - (1 - y_k^4)^{1/4}}{1 + (1 - y_k^4)^{1/4}} \\ a_{k+1} &= a_k (1 + y_{k+1})^4 - 2^{2k+3} y_{k+1} (1 + y_{k+1} + y_{k+1}^2) \end{aligned}$$

Then a_k converges quartically to $1/\pi$ — each iteration approximately quadruples the number of correct digits. This particular algorithm, together with the Salamin–Brent scheme, has been employed by Yasumasa Kanada of the University of Tokyo in several computations of π over the past 15 years or so (although evidently not in the most recent computation).

A summary of π computations is given in Table 1. You inquired about "gaps" in the table. Indeed, there is a "gap" of sorts between 1967 and 1982. As far as I can tell, this "jump" reflects mainly the introduction of FFT-based arithmetic, as well as the usage of the Salmin-Brent π algorithm. I haven't personally checked whether these computations track Moore's Law accurately, but the steady progress since 1982 at least seems in keeping with Moore's Law during this time.

Calculating Individual Digits

Before discussing Kanada's work more, I might mention some work of my own, since it connects with something I will mention later. In 1996, Peter Borwein, Simon Plouffe and I found a way to calculate *individual* digits of π. In particular, our scheme permits one to calculate a segment of hexadecimal (base 16) or binary digits beginning at the n-th position, without having to calculate any of the first $n-1$ digits, using a very simple algorithm that requires only a very small amount of computer memory, and does not require multi-precision arithmetic. Using this algorithm, for example, the one millionth hexadecimal digit (or the four millionth binary digit) of π can be computed in less than a minute on a 2001-era personal computer. This scheme is based on the following formula for π (which was discovered by a computer program running Ferguson's PSLQ algorithm):

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

Moving from this formula to the individual digit computation scheme only requires two or three lines of math, but I won't include it here. Several persons have actually done some computations of this sort, as summarized in Table 2. The last listed computation was done by Colin Percival, an undergraduate at Simon Fraser University, utilizing a worldwide network of over 1700 computers.

Recently Richard Crandall and I showed that the existence of this particular formula, plus numerous similar formulas that have been found by PSLQ computer searches since 1996, has significant implications for the question of whether and why the digits of π are "random." See the recent article in *Science* for some details [2]. More precisely, we say that a constant is *normal* if each digit appears, in the limit, one-tenth of the time, that each pair of digits appears, in the limit, one one-hundredth of the time, and so on. On one hand it is easy to show that "almost all" real numbers are normal, but on the other hand none of the fundamental constants of mathematics has ever been proven normal, not to any number base. Trying to understanding why π and other math constants appear to be normal is a centuries-old unsolved question in mathematics.

What Crandall and I demonstrated in our 2001 paper is that the question of the normality of π base 16 (or base 2), as well as the normality of numerous other mathematical constants, can be reduced to a plausible but unproven conjecture in the arena of chaotic sequences. This work is a direct outgrowth of the discovery of the new π formula above. More recently, Crandall and I were actually able to prove normality (not conditional on any unproven conjecture) for a certain infinite class of real numbers (sadly not including π). In any event, there is some hope

Name	Year	Correct Digits
Archimedes	250? BCE	3
Ptolemy	150	3
Liu Hui	263	5
Tsu Ch'ung Chi	480?	7
Al-Kashi	1429	14
Romanus	1593	15
Van Ceulen	1615	35
Sharp	1699	71
Machin	1706	100
Strassnitzky and Dase	1844	200
Rutherford	1853	440
Shanks	1874	527
Reitwiesner et al. (ENIAC)	1949	2,037
Genuys	1958	10,000
Shanks and Wrench	1961	100,265
Guilloud and Bouyer	1973	1,001,250
Miyoshi and Kanada	1981	2,000,036
Kanada, Yoshino and Tamura	1982	16,777,206
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 et. al	Jan. 1987	134,217,700
Kanada and Tamura	Jan. 1988	201,326,551
Chudnovskys	May 1989	480,000,000
Kanada and Tamura	Jul. 1989	536,870,898
Kanada and Tamura	Nov. 1989	1,073,741,799
Chudnovskys	Aug. 1991	2,260,000,000
Chudnovskys	May 1994	4,044,000,000
Kanada and Takahashi	Oct. 1995	6,442,450,938
Kanada and Takahashi	Jul. 1997	51,539,600,000
Kanada and Takahashi	Sep. 1999	206,158,430,000
Kanada, Ushiro, Kuroda	Dec. 2002	1,241,100,000,000

Table 1: Chronicle of π Calculations

Position	Hex Digits Beginning At This Position
10^6	26C65E52CB4593
10^7	17AF5863EFED8D
10^8	ECB840E21926EC
10^9	85895585A0428B
10^{10}	921C73C6838FB2
10^{11}	9C381872D27596
1.25×10^{12}	07E45733CC790B
2.5×10^{14}	E6216B069CB6C1

Table 2: Computed Hexadecimal Digits of π

now that this research will lead to the long-sought explanation of why the digits of π (at least the binary or hexadecimal digits) appear "random."

Kanada's Latest Computation

Recently Kanada, with a team consisting of Y. Ushiro of Hitachi, H. Kuroda and M. Kudoh of the University of Tokyo, and the assistance of nine others from Hitachi, computed π to over 1.24 *trillion* decimal digits. Kanada and his team first computed π in hexadecimal (base 16) to 1,030,700,000,000 places, using the following two arctangent relations for π:

$$\begin{aligned} \pi &= 48 \arctan \frac{1}{49} + 128 \arctan \frac{1}{57} - 20 \arctan \frac{1}{239} + 48 \arctan \frac{1}{110443} \\ \pi &= 176 \arctan \frac{1}{57} + 28 \arctan \frac{1}{239} - 48 \arctan \frac{1}{682} + 96 \arctan \frac{1}{12943}. \end{aligned} \tag{1}$$

The first formula was found in 1982 by K. Takano, a high school teacher and song writer. The second formula was found by F. C. W. Störmer in 1896.

Kanada and his team evaluated these formulas using a scheme analogous to that employed by Gosper and the Chudnovskys, in that they were able to avoid explicitly storing the multiprecision numbers involved. This resulted in a scheme that is roughly competitive in efficiency compared to the Salamin-Brent and Borwein quartic algorithms they had previously used, yet with a significantly lower total memory requirement. In particular, they were able to perform their latest computation on a system with 1 Tbyte (10^{12} bytes) main memory, the same as with their previous computation, yet obtain six times as many digits.

After Kanada and his team verified that the hexadecimal digit strings produced by these two computations were in agreement, they performed an additional check by directly computing 20 hexadecimal digits beginning at position 1,000,000,000,001. This calculation employed the algorithm that described above for computing individual hexadecimal digits of π, and required 21 hours run time, much less than the time required for the first step. The result of this calculation, `B4466E8D21 5388C4E014`, perfectly agreed with the corresponding digits produced by the two arctan formulas. At this point they converted their hexadecimal value of π to decimal, and converted back to hexadecimal as a check. These conversions employed a numerical approach similar to that used in the main and verification calculations. The entire computation, including hexadecimal and decimal evaluations and checks, required roughly 600 hours run time on their

64-node Hitachi parallel supercomputer. The main segment of the computation ran at nearly 1 Tflop/s (i.e., one trillion floating-point operations per second), although this performance rate was slightly lower than the rate of their previous calculation of 206 billion digits. Full details will appear in an upcoming paper..

According to Kanada, the ten decimal digits ending in position one trillion are 6680122702, while the ten hexadecimal digits ending in position one trillion are `3F89341CD5`. Some data on the frequencies of digits in π, based on Kanada's computations, are available at Kanada's website:

`http://www.super-computing.org`

Motivation

One final question is what is the motivation behind these modern comptuations of π, given that questions such as the irrationality and transcendence of π were settled more than 100 years ago. One objective is to demonstrate that these recently-discovered algorithms really do work at a massive scale (this is Kanada's principal motivation), as well as the raw challenge of harnessing the stupendous power now available in modern computer systems for this classical computational problem. I should add that programming such calculations is definitely not trivial, especially on large, distributed memory computer systems. Kanada mentions that his team has worked for five years on the program they used, but I am sure he means here is five years of on-and-off-again programing work by various people, while mainly doing other "official" assignments at the computer center.

There have been some practical spin-off benefits from these efforts through the years. For example, some new techniques for performing the fast Fourier transform (FFT), had their roots in attempts to accelerate computations of π.

Beyond purely practical considerations, there is of course continuing interest in the fundamental question of the normality of π, as I mentioned above. Kanada has performed detailed statistical analyses of his computed results of π in the past, and is commencing to perform such analyses on his newly computed digit streams, to see if there are any statistical abnormalities that suggest π is not normal. One unique aspect of Kanada's latest computation is that he calculated hexadecimal digits as well as decimal digits. I for one am looking forward to seeing the results of statistical analyses on these hexadecimal digits, in part because the new normality and individual digit calculation properties apply to hexadecimal digits, not decimal digits.

Certainly the normality of π is not going to be formally settled by such a computation (except perhaps in the exceedingly unlikely scenario that a significant statistical anomaly is found), nor is it likely that this computation will provide major insights leading to a formal proof. In other words, being able to perform statistical analyses on the digits is a nice by-product of this computation, but it is hardly by itself a realistic justification for expending this much computer time.

The computer feat is the main achievement...

References

References

[1] David H. Bailey, Jonathan M. Borwein, Peter B. Borwein and Simon Plouffe, "The Quest for Pi," *Mathematical Intelligencer*, vol. 19, no. 1 (January 1997), pg. 50–57.

[2] , Charles Seife, "Randomly Distributed Slices of Pi," *Science*, 3 Aug 2001, pg. 793; the text of this article is available from the URL
`http://sciencenow.sciencemag.org/cgi/content/full/2001/727/1`