A More Efficient Algorithm to Compute π

Ronald S. Remmel

December 8, 2014

Abstract

The calculation of more digits of π, an irrational number, has fascinated mathematicians since antiquity. The current record is 12 trillion digits (12×10^{12}) calculated by Yee and Kondo on a PC; they used the Chudnowsky power-series formula. We here present the new half-angle algorithm, which is about 76,000 times more efficient. First it repeatedly computes the tangent of half the angle, starting with $\tan(\pi/4) = 1$, until the tangent becomes very small. Then it uses the power series expansion for arctan to compute π. Because the tangent is so small, the series converges very quickly. This method is more efficient than any other power series method, but slower than iterative methods.

1 Introduction

The calculation of the exact value of π has fascinated mathematicians since antiquity. [1] The ancients discovered that 22/7 was a good approximation. Archimedes [1] around 250 B.C. approximated a circle with 96-sided polygrams inside and outside, proving that $223/71 < \pi < 22/7$. It was proved that π, like $\sqrt{2}$, is an irrational number, requiring an infinite number of decimal digits. In the 16th and 17th centuries infinite series were used to approximate π. Many series are based upon the power series for arc tangent discovered by Madhava, Gregory, and Leibniz:[2]

$$\arctan(x) = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{2i+1} \tag{1}$$

Because $\tan(\pi/4) = 1$, we get the series:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} ... \tag{2}$$

Unfortunately this series converges too slowly to be useful for computing π. In 1706 Machin [3] produced an algorithm which converged much more rapidly:

$$\frac{\pi}{4} = 4 \arctan \frac{1}{5} - \arctan \frac{1}{239} \tag{3}$$

Machin computed π to 100 digits with this formula.

Several faster converging Machin-like formulas were discovered,[3] such as:

$$\frac{\pi}{4} = 183 \arctan \frac{1}{239} + 32 \arctan \frac{1}{1023} - 68 \arctan \frac{1}{5832} \tag{4}$$

$$+12 \arctan \frac{1}{110443} - 12 \arctan \frac{1}{4841182} - 100 \arctan \frac{1}{6826318} \tag{5}$$

Through the years mathematicians calculated to several hundred digits using Machin-like formulas. Finally Wrench and Smith in 1949 [1] achieved 1120 digits using a desk calculator. Then the world's first electronic computer–ENIAC–in 1949 calculated π to 2037 digits [1]; Reitwiesner and von Neumann used 70 hours of computer time! Machin-like formulas and computers were used to break the record repeatedly until 1,000,000 digits were reached in 1973.

A more efficient algorithm for π was devised by Ramanujan: [1]

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{i=0}^{\infty} \frac{(4i)!(1103 + 26390i)}{(i!)^4 (396)^{4i}} \tag{6}$$

Chudnovsky and Chudnovsky developed an even more efficient formula [4]:

$$\frac{1}{\pi} = \frac{12}{640320^{3/2}} \sum_{i=0}^{\infty} \frac{(6i)!(13591409 + 545140134i)}{(3i)!(i!)^3 (-640320)^{3i}} \tag{7}$$

This formula was used to compute π to 1 billion digits (10^9) by the Chudnovsky brothers in 1989, to 2.7 trillion digits (2.7×10^{12}) by Bellard in 2009 [5], and to 10 trillion digits (10^{13}) in 2011 by Yee and Kondo [6].

A similar but more advanced formula was developed by Borwein and Borwein in 1993 [7].

2 Methods

We now present a more efficient algorithm. The power series for arc tangent (eq. 1) converges more rapidly the smaller x is, so we make x as small as possible through repeated use of the half-angle formula for tangent: [8]

$$\tan \frac{\theta}{2} = \frac{\sin \theta}{1 + \cos \theta} = \frac{\tan \theta}{\frac{1}{\cos \theta} + 1} = \frac{\tan \theta}{\frac{\sqrt{\cos^2 \theta + \sin^2 \theta}}{\cos \theta} + 1} = \frac{\tan \theta}{\sqrt{1 + \tan^2 \theta} + 1} \tag{8}$$

We have divided the numerator and denominator of the second term by $\cos \theta$ to get the third term. Then we replaced the 1 by $\sqrt{\cos^2 \theta + \sin^2 \theta}$ to get the fourth term. Finally we divided the square root by $\cos \theta$ to get the final term.

We can thus define the half-angle operator which takes the tangent x of any angle and computes the tangent of half that angle:

$$H \bullet x = \frac{x}{\sqrt{1+x^2}+1} \tag{9}$$

The algorithm is to start with:

$$\tan\frac{\pi}{4} = 1 = x \tag{10}$$

and apply the half-angle operator k times to get:

$$\tan\frac{\pi}{2^{k+2}} = H^k \bullet 1 \tag{11}$$

Then the arctan power series (eq. 1) is used to calculate $\pi/2^{k+2}$. The series will rapidly converge because $H^k \bullet 1$ is so small.

π accurate to 30,000,000 digits was computed on a Hewlett Packard PC with an AMD A6-3620 APU processor. Linux Fedora 20 (64-bit) and the g++ compiler were used to program in C++.

The *MPFR* multiprecision floating reliable functions [9] were used for all calculations. The citation indicates where to download the MPFR instruction manual and the programs. The copiously-commented program "piMPFR.cpp" is available on the Internet, [10] and contains instructions to download, install, and run Linux, the g++ compiler, and MPFR.

How many terms p are needed in the power series (eq. 1)? Because for small angles the arctangent of an angle is approximately equal to that angle, $x \approx \pi/2^{k+2}$. To achieve N decimal digits of accuracy:

$$10^{-N} \approx \frac{(\pi/2^{k+2})^{2p+1}}{2p+1} \tag{12}$$

By taking $\log_{10}$ of both sides, neglecting small terms because k and p are large, and solving for p, we get:

$$p \approx \frac{N}{2k\log_{10}2} \tag{13}$$

Note that p goes inversely with k; we can reduce the number of terms needed in the series simply by doing more half-angles! Define:

t_k = time to compute one half-angle = 10.79 s.

t_p = time to compute one term in the power series = 1.791 s.

$$T = total\ time = kt_k + pt_p = kt_k + t_p\frac{N}{2klog_{10}2} \tag{14}$$

T was minimized by setting the derivative with respect to k to zero, giving for N=30,000,000:

$$k = \sqrt{\frac{t_pN}{t_k 2\log_{10}2}} \approx 2882 \tag{15}$$

Note that the number of calculations required increases only as $\sqrt{N}$. Make k=2900 to be conservative. Then the equation for p gives p = 17,182; to be safe set p = 17,300.

3 Results

The program [10] required 9.5 hr to compute the half-angles, and 8.8 hr to compute the power series. The resulting file *piMPFR.out*, posted on the Internet, [11] agrees with the 100,000 digits computed by Huberty et al. [12] to within the last 3 digits.

The resulting file *piMPFR.hex* in hexadecimal has also been posted [13]; this file can easily be converted to a binary file if desired.

4 Discussion

We can estimate the efficiency of an algorithm for computing π by estimating the number of additional decimal digits of accuracy achieved, D, for each additional term added to the series. We compute the Chudnovsky formula (eq. 7) as an example; divide the (i+1)th term by the ith term:

$$r = \frac{(i+1)st \quad term}{ith \quad term} = \frac{\frac{(6i+6)!(13591409+545140134(i+1))}{(3i+3)!((i+1)!)^3(-640320)^{3i+3}}}{\frac{(6i)!(13591409+545140134i)}{(3i)!(i!)^3(-640320)^{3i}}} \tag{16}$$

Because i becomes very large, the number 13591409 can be neglected. Also ignore the minus sign. Terms in the numerator and denominator cancel:

$$\approx \frac{(6i+6)(6i+5)(6i+4)(6i+3)(6i+2)(6i+1)}{(3i+3)(3i+2)(3i+1)(i+1)^3 640320^3} \tag{17}$$

Because i becomes very large, the small integers can be neglected:

$$\approx \frac{6^6}{3^3 \times 640320^3} = 6.58 \times 10^{-15} \tag{18}$$

Thus the Chudnovsky formula gives about $D = -\log_{10} r = 14.2$ decimal digits more accuracy per term.

Algorithm	D, digits/term
Machin, eq. 3	1.4
Machin-like, eq. 4	4.8
Ramanujan, eq. 6	8.0
Chudnovsky, eq. 7	14.2
Borwein [7]	50.6
Half-angle	$2(k+2)\log_{10} 2$

Thus with the half-angle algorithm, the number of digits accuracy gained per term can be made as large as desired by increasing k, the number of times the half-angle was computed.

The efficiency of the half-angle algorithm is derived as follows. Because for small angles, $\tan\theta \approx \theta$, x in the arctan power series is $\approx \pi/2^{k+2}$. Thus:

$$r = \frac{(\pi/2^{k+2})^{2p+3}/(2p+3)}{(\pi/2^{k+2})^{2p+1}/(2p+1)} \approx (\pi/2^{k+2})^2 \Rightarrow D = -\log_{10} r \approx 2(k+2)\log 2 \quad (19)$$

Let us compare the efficiency of the various methods for computing π to 10 trillion digits, the current record.

Algorithm	Terms needed
Machin, eq. 3	7,200,000,000,000
Machin-like, eq. 4	2,100,000,000,000
Ramanujan, eq. 6	1,250,000,000,000
Chudnovsky, eq. 7	700,000,000,000
Borwein [7]	200,000,000,000
Half-angle	9,113,000

The half-angle estimate was computed by using eqs. 15 and 13, and by assuming that computing a half-angle takes about 5× as long as computing a term in the power series. This beats the Chudnovsky algorithm by 76,000 times.

Iterative methods are still faster, but might require more memory. The Gauss-Legendre algorithm, as modified by Brent and independently by Salamin, [14] converges quadratically. There are other iterative algorithms which converge at least quadratically. [7]

In practice for calculations to 10^{13} digits, the program must store some numbers on a hard drive because of a lack of random-access memory. The GNU MP multiprecision library does not work on disk drives. Disk drives fail and power goes out during year-long calculations. The algorithm chosen depends on expediency.

References

[1] Pi, http://en.wikipedia.org/wiki/Pi

[2] Madhava series, http://en.wikipedia.org/wiki/Madhava_series

[3] Machin-like formula, http://en.wikipedia.org/wiki/Machin-like_formula

[4] D.V. Chudnovsky and G.V. Chudnovsky, Approximations and complex multiplication according to Ramanajan, *Ramanujan Revisited. Proceedings of the Centenary Conference, University of Illinois at Urbana-Champaign, June 1-5, 1987*, Andrews et al., ed., Academic Press, Boston, 1988, pp. 375-472

[5] F. Bellard, Computation of 2700 billion decimal digits of Pi using a desktop computer, 2010, http://bellard.org/pi/pi2700e9/pipcrecord.pdf

[6] A.J. Yee and S. Kondo, Round 2...10 trillion digits of pi, 2013, http://numberworld.org/misc_runs/pi-10t/details.html

[7] Borwein's algorithm, http://en.wikipedia.org/wiki/Borwein's_algorithm

[8] Proofs of trigonometric identities, http://en.wikipedia.org/wiki/Proofs_of_trigonometric_identities#Half-angle_identities

[9] The GNU MPFR Library, http://www.mpfr.org

[10] http://angelfire.com/scifi2/remmel.labs/piMPFR.cpp

[11] 30,000,000 digits of π, http://angelfire.com/scifi2/remmel.labs/piMPFR.out

[12] M. Huberty et al., 100,000 digits of π, http://www.geom.uiuc.edu/~huberty/math5337/groupe/digits.html

[13] 24,914,400 hexadecimal digits of π, http://angelfire.com/scifi2/remmel.labs/piMPFR.hex

[14] Gauss-Legendre algorithm http://en.wikipedia.org/wiki/Gauss-Legendre_algorithm

Biography: Ronald S. Remmel] (remmellabs@juno.com www.remmellabs.com) received his BS in physics from Caltech, and his PhD in physics from Princeton. He was on the faculties of the Univ. of Arkansas for Medical Science and Boston Univ. He has published 26 scientific papers, 1 computer book, 4 medical course manuals, and 8 science fiction novels. In 1991 he founded Remmel Labs, which manufactures eye movement monitors for visual and oculomotor experiments.

Remmel Labs, 1811 Parkfair Ct., Katy, TX 77450 USA
remmellabs@juno.com