



2023

Euler Found the First Binary Digit Extraction Formula for π in 1779

Nick Craig-Wood

None, nick@craig-wood.com

Follow this and additional works at: <https://scholarlycommons.pacific.edu/euleriana>



Part of the [Mathematics Commons](#)



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#)

Recommended Citation

Craig-Wood, Nick (2023) "Euler Found the First Binary Digit Extraction Formula for π in 1779," *Euleriana*: 3(1), p. 23, Article 3.

DOI: <https://doi.org/10.56031/2693-9908.1049>

Available at: <https://scholarlycommons.pacific.edu/euleriana/vol3/iss1/3>

This Articles & Notes is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in Euleriana by an authorized editor of Scholarly Commons. For more information, please contact mgibney@pacific.edu.

Euler found the first binary digit extraction formula for π in 1779

Nick Craig-Wood
nick@craig-wood.com

Abstract

In 1779 Euler discovered two formulas for π which can be used to calculate any binary digit of π without calculating the previous digits. Up until now it was believed that the first formula with the correct properties (known as a BBP-type formula) for this calculation was published by Bailey, Borwein and Plouffe in 1997.

1 Introduction

In 1997 Bailey, Borwein and Plouffe published their paper [3] demonstrating how to calculate any binary or hexadecimal digit of π without calculating the previous digits. Remarkably this algorithm needs constant space and runs in $O(n \log n)$ time making it extremely practical.

The calculation revolves about this equation which was discovered using a computer-aided search using the integer relation algorithm PSLQ [6].

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

This style of equation is now known as a BBP-type formula after its inventors Bailey, Borwein and Plouffe.

What the authors didn't know is that Leonhard Euler (1707-1783) discovered two BBP-type formulas which would have worked just as well as the formula for which they spent a lot of computer time searching.

Euler's 1779 paper "De novo genere serierum rationalium et valde convergentium quibus ratio peripheriae ad diametrum exprimi potest" [5] (E706) (as translated by Jordan Bell with the English title "On a new type of rational and highly convergent series, by which the ratio of the circumference to the diameter is able to be expressed") contains the formulas. Those who wish to read more about the history of this paper should consult Ed Sandifer's article [8] from his "How Euler Did It" series.

It isn't immediately obvious what the BBP-type formula is as Euler left it in expanded form:

$$\pi = \begin{cases} 2(1 - \frac{1}{5} \cdot \frac{1}{4} + \frac{1}{9} \cdot \frac{1}{4^2} - \frac{1}{13} \cdot \frac{1}{4^3} + \frac{1}{17} \cdot \frac{1}{4^4} - etc.) \\ +1(1 - \frac{1}{3} \cdot \frac{1}{4} + \frac{1}{5} \cdot \frac{1}{4^2} - \frac{1}{7} \cdot \frac{1}{4^3} + \frac{1}{9} \cdot \frac{1}{4^4} - etc.) \\ +1(\frac{1}{3} - \frac{1}{7} \cdot \frac{1}{4} + \frac{1}{11} \cdot \frac{1}{4^2} - \frac{1}{15} \cdot \frac{1}{4^3} + \frac{1}{19} \cdot \frac{1}{4^4} - etc.) \end{cases}$$

But in more compact notation it is this, where the similarities to equation (1) are immediately apparent:

$$\pi = \sum_{n=0}^{\infty} \left[\frac{(-1)^n}{4^n} \left(\frac{2}{4n+1} + \frac{2}{4n+2} + \frac{1}{4n+3} \right) \right]. \tag{2}$$

This was discovered independently by Adamchik and Wagon [1] in 1997. Euler also produced this formula ^a:

$$\pi = \begin{cases} 2(1 - \frac{1}{5} \cdot \frac{1}{64} + \frac{1}{9} \cdot \frac{1}{64^2} - \frac{1}{13} \cdot \frac{1}{64^3} - etc.) \\ +\frac{1}{2}(1 - \frac{1}{3} \cdot \frac{1}{64} + \frac{1}{5} \cdot \frac{1}{64^2} - \frac{1}{7} \cdot \frac{1}{64^3} - etc.) \\ +\frac{1}{4}(\frac{1}{3} - \frac{1}{7} \cdot \frac{1}{64} + \frac{1}{11} \cdot \frac{1}{64^2} - \frac{1}{15} \cdot \frac{1}{64^3} - etc.) \\ +\frac{1}{2}(1 - \frac{1}{5} \cdot \frac{1}{1024} + \frac{1}{9} \cdot \frac{1}{1024^2} - \frac{1}{13} \cdot \frac{1}{1024^3} - etc.) \\ +\frac{1}{16}(1 - \frac{1}{3} \cdot \frac{1}{1024} + \frac{1}{5} \cdot \frac{1}{1024^2} - \frac{1}{7} \cdot \frac{1}{1024^3} - etc.) \\ +\frac{1}{64}(\frac{1}{3} - \frac{1}{7} \cdot \frac{1}{1024} + \frac{1}{11} \cdot \frac{1}{1024^2} - \frac{1}{15} \cdot \frac{1}{1024^3} - etc.) \end{cases}$$

With a bit of effort (as will be described below) this can be rewritten as:

$$\pi = \frac{1}{2^{26}} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^{30n}} \left[\begin{aligned} & \frac{2^{27}}{20n+1} + \frac{2^{25}}{12n+1} + \frac{2^{25}}{10n+1} + \frac{2^{24}}{20n+3} + \frac{2^{22}}{6n+1} \\ & - \frac{2^{20}}{60n+15} - \frac{2^{19}}{10n+3} - \frac{2^{18}}{20n+7} - \frac{2^{15}}{12n+5} \\ & + \frac{2^{15}}{20n+9} + \frac{2^{12}}{30n+15} + \frac{2^{12}}{20n+11} - \frac{2^{10}}{12n+7} \\ & - \frac{2^9}{20n+13} - \frac{2^7}{10n+7} - \frac{2^5}{60n+45} + \frac{2^3}{20n+17} \\ & + \frac{2^2}{6n+5} + \frac{2}{10n+9} + \frac{1}{12n+11} + \frac{1}{20n+19} \end{aligned} \right]. \tag{3}$$

^aNote that this equation is incorrect in the original paper and the translation - the bottom left term is printed as 1 whereas it should be $\frac{1}{3}$ as shown here.

This is clearly a BBP-type formula and it was discovered independently by Pschill [2] in 1999.

2 Euler's Derivation of Equation (2)

Starting from the observation $v^4 + 4 = (v^2 + 2v + 2)(v^2 - 2v + 2)$, Euler observed:

$$\int_0^x \frac{v^2 + 2v + 2}{v^4 + 4} dv = \int_0^x \frac{dv}{v^2 - 2v + 2} \quad (4)$$

$$= -\arctan(1 - x) + \arctan 1 \quad (5)$$

$$= \arctan \frac{x}{2 - x} \quad (6)$$

Euler then went to work on the left hand side of this equation by expanding $\frac{v^k}{v^4 + 4}$, where k is an integer, as an infinite series by polynomial division.

$$\frac{v^k}{v^4 + 4} = \sum_{n=0}^{\infty} (-1)^n \frac{v^{4n+k}}{4^{n+1}} \quad (7)$$

Integrating this we get

$$\int_0^x \frac{v^k}{v^4 + 4} dv = \int_0^x \sum_{n=0}^{\infty} (-1)^n \frac{v^{4n+k}}{4^{n+1}} dv \quad (8)$$

$$= \sum_{n=0}^{\infty} (-1)^n \frac{x^{4n+k+1}}{(4n + k + 1)4^{n+1}} \quad (9)$$

We can now rewrite the LHS of equation (6) as

$$\int_0^x \frac{v^2 + 2v + 2}{v^4 + 4} dv = \int_0^x \frac{v^2}{v^4 + 4} dv + \int_0^x \frac{2v}{v^4 + 4} dv + \int_0^x \frac{2}{v^4 + 4} dv \quad (10)$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{4^{n+1}} \left[\frac{x^{4n+3}}{4n + 3} + \frac{2x^{4n+2}}{4n + 2} + \frac{2x^{4n+1}}{4n + 1} \right] \quad (11)$$

Equating equation (11) and equation (6) Euler produced this infinite series for $\arctan \frac{x}{2-x}$:

$$\arctan \frac{x}{2-x} = \sum_{n=0}^{\infty} \frac{(-1)^n}{4^{n+1}} \left[\frac{2x^{4n+1}}{4n+1} + \frac{2x^{4n+2}}{4n+2} + \frac{x^{4n+3}}{4n+3} \right] \quad (12)$$

Euler then substituted $x = 1$ to produce $\arctan 1 = \frac{\pi}{4}$ to create the BBP-type equation (2) directly.

3 Euler's Derivation of Equation (3)

Euler obviously realised that equation (12) had more to give in terms of π generating sequences. Equation (2) generates roughly 4 bits of π or 1.2 decimal digits per iteration. To do better he took this identity that was first discovered by Hutton [2] in 1776:

$$\pi = 8 \arctan \frac{1}{3} + 4 \arctan \frac{1}{7}. \quad (13)$$

He then used equation (12) with $x = \frac{1}{2}$ and $x = \frac{1}{4}$ to give series for $\arctan \frac{1}{3}$ and $\arctan \frac{1}{7}$, respectively. Combining them Euler got:

$$\begin{aligned} \pi &= \frac{1}{4} \sum_{n=0}^{\infty} \frac{(-1)^n}{64^n} \left[\frac{8}{4n+1} + \frac{4}{4n+2} + \frac{1}{4n+3} \right] \\ &+ \frac{1}{64} \sum_{n=0}^{\infty} \frac{(-1)^n}{1024^n} \left[\frac{32}{4n+1} + \frac{8}{4n+2} + \frac{1}{4n+3} \right]. \end{aligned} \quad (14)$$

This was as far as Euler went. While we can clearly see it is a BBP-type formula, it isn't yet in the form we expect to see.

To write this as a single BBP-type formula, we can collect groups of terms (five for the first sum, three for the second) to get a common multiplier of $\frac{(-1)^n}{2^{30n}}$.

$$\pi = \frac{1}{4} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^{30n}} \left[\begin{array}{l} \frac{8}{20n+1} + \frac{4}{20n+2} + \frac{1}{20n+3} \\ - \frac{8.2^{-6}}{20n+5} - \frac{4.2^{-6}}{20n+6} - \frac{1.2^{-6}}{20n+7} \\ + \frac{8.2^{-12}}{20n+9} + \frac{4.2^{-12}}{20n+10} + \frac{1.2^{-12}}{20n+11} \\ - \frac{8.2^{-18}}{20n+13} - \frac{4.2^{-18}}{20n+14} - \frac{1.2^{-18}}{20n+15} \\ + \frac{8.2^{-24}}{20n+17} + \frac{4.2^{-24}}{20n+18} + \frac{1.2^{-24}}{20n+19} \end{array} \right] + \frac{1}{64} \sum_{n=0}^{\infty} \frac{(-1)^n}{2^{30n}} \left[\begin{array}{l} \frac{32}{12n+1} + \frac{8}{12n+2} + \frac{1}{12n+3} \\ - \frac{32.2^{-10}}{12n+5} - \frac{8.2^{-10}}{12n+6} - \frac{1.2^{-10}}{12n+7} \\ + \frac{32.2^{-20}}{12n+9} + \frac{8.2^{-20}}{12n+10} + \frac{1.2^{-20}}{12n+11} \end{array} \right] \quad (15)$$

After much simplification, this becomes equation (3).

This formula calculates 30 bits of π per iteration or about 9 decimal places. For binary digit extraction purposes it is about 1.42 times faster than equation (1) as it calculates $30/21 = 1.42$ bits of π per division as compared to $4/4 = 1$ bit of π per division for the original equation.

4 Calculating binary digits of π

The secret to calculating the m -th binary digit of π without calculating the previous digits is to multiply equation (2) by 2^m and to calculate the fractional parts only which we will denote as $(\text{mod } 1)$ arithmetic. Printing this fractional result as a binary or hexadecimal floating point number will reveal the m -th binary digit of π and some following digits.

$$2^m \pi = \left[\begin{array}{l} \sum_{n=0}^{\lfloor m/2 \rfloor} \left[(-1)^n 2^{m-2n} \left(\frac{2}{4n+1} + \frac{2}{4n+2} + \frac{1}{4n+3} \right) \right] \\ + \sum_{n=\lfloor m/2 \rfloor + 1}^{\infty} \left[\frac{(-1)^n}{2^{2n-m}} \left(\frac{2}{4n+1} + \frac{2}{4n+2} + \frac{1}{4n+3} \right) \right] \end{array} \right] \pmod{1} \quad (16)$$

The second half of this equation is easy to calculate as the absolute value of all terms is less than 1 and very quickly these will become so small that we don't need to worry about them. In general, $\lfloor p_f/2 \rfloor$ terms will be sufficient for a floating point number with precision p_f .

The first half looks more difficult to calculate at first glance due to the large powers involved; however we can use modular arithmetic to calculate this as:

$$2^m \pi \approx \left[\sum_{n=0}^{\lfloor m/2 \rfloor} (-1)^n \left(\frac{2^{m-2n+1} \pmod{4n+1}}{4n+1} + \frac{2^{m-2n+1} \pmod{4n+2}}{4n+2} + \frac{2^{m-2n} \pmod{4n+3}}{4n+3} \right) \right] + \sum_{n=\lfloor m/2 \rfloor+1}^{\lfloor m/2 \rfloor + \lfloor p_f/2 \rfloor} \left[\frac{(-1)^n}{2^{2n-m}} \left(\frac{2}{4n+1} + \frac{2}{4n+2} + \frac{1}{4n+3} \right) \right] \pmod{1}. \tag{17}$$

All parts of this equation are now computable using fixed precision floating point numbers of precision p_f bits and integers of precision p_i bits; $2^x \pmod{k}$ can easily be computed using only $O(\log n)$ operations [7] using an integer of sufficient size to hold k .

There are two limits to the accuracy of this equation. Firstly we need the maximum denominator to fit into p_i bits this means that $4(m/2) + 3 < 2^{p_i}$ so $m < 2^{p_i-1} - 1.5$. Secondly each time we add a floating point number we will, on average, add half a bit of inaccuracy to the result. This equation needs 3 additions per iteration so our final computed result will only have accuracy approximately $p_f - \log_2(3(m/2)/2) = p_f - \log_2(3m/4)$ bits.

5 Computation

A program to calculate hex digits of π was written based on equation (1), (2) and (3) using 64 bit integers ($p_i = 64$) and double precision floating point ($p_f = 53$). Table 1 shows the result of the computation on the author's laptop.

It can be seen in the table that Equation (3) is 1.42 times faster than Equation (1) as expected, but Equation (2) is 1.5 times slower.

The program which generated this table can be downloaded from the website for this article [4].

| Digit | Hex Result | Equation (1) | Equation (2) | Equation (3) |
|-----------|-------------|--------------|--------------|--------------|
| 10^1 | 5A308D31319 | 31.1 μ s | 33.0 μ s | 33.0 μ s |
| 10^2 | C29B7C97C50 | 37.6 μ s | 61.3 μ s | 33.9 μ s |
| 10^3 | 349F1C09B0 | 231 μ s | 342 μ s | 127 μ s |
| 10^4 | 68AC8FCFB | 2.23ms | 3.36ms | 1.45ms |
| 10^5 | 535EA16C | 27.0ms | 21.5ms | 17.0ms |
| 10^6 | 26C65E52 | 57.8ms | 83.0ms | 162.5ms |
| 10^7 | 17AF586 | 476ms | 707ms | 392ms |
| 10^8 | ECB840 | 5.95s | 8.93s | 4.07s |
| 10^9 | 85895 | 66.7s | 100.5s | 44.6s |
| 10^{10} | 921C | 12m56s | 19m38s | 8m53s |
| 10^{11} | C9C | 2h28m | 3h43m | 1h37m |

Table 1: The 10^m -th hex digits of π and the calculation time for each equation.

6 Conclusion

Euler noted about equation (3), “This occurs as most noteworthy, because all these series proceed only by powers of two” and it is indeed this property which enables its use as a binary digit extraction formula for π . Euler did not make the leap to note that this formula could be used for binary digit extraction — that was Bailey, Borwein and Plouffe’s stroke of genius — but amazingly he did find two formulas of the right form 200 years earlier.

References

- [1] Victor Adamchik and Stan Wagon. A simple formula for π . *The American Mathematical Monthly*, 104(9):852–855, Nov 1997.
- [2] Jörg Arndt and Christoph Haenel. *Pi — Unleashed*, chapter 16. π Formula Collection, pages 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [3] David Bailey, Peter Borwein, and Simon Plouffe. On the rapid computation of various polylogarithmic constants. *Mathematics of Computation*, 66(218):903–913, 1997.

- [4] Nick Craig-Wood. Programs to calculate π using BBP-type formulae, 2023. Available at https://github.com/ncw/euler_bbp.
- [5] Leonhard Euler. De novo genere serierum rationalium et valde convergentium quibus ratio peripheriae ad diametrum exprimi potest (E706). *Nova acta academiae scientiarum Petropolitanae 11 (1793), 1798, pp. 150-154, 169. Reprinted in Opera Omnia, Series I vol . 16, pp. 21-27.* Original text available online, along with an English translation by Jordan Bell, at <https://scholarlycommons.pacific.edu/euler-works/706/>.
- [6] Helaman R. P. Ferguson and David H. Bailey. A polynomial time, numerically stable integer relation algorithm. Technical Report RNR-91-032, RNR, 1992.
- [7] Donald Ervin Knuth. *The art of computer programming. 2: Seminumerical algorithms.* Addison-Wesley, Reading, Mass, 2. ed., 24. [print.]-1995 edition, 1995.
- [8] Ed Sandifer. How Euler did it. *MAA Online*, 2009. Available at <http://eulerarchive.maa.org/hedi/HEDI-2009-02.pdf>.