

# Decimal Digit Extraction for $\pi$ via Machin’s Formula: Series Splitting and Borrow-Tracking Methods

Joseph Babcanec  
Benedict College  
`joseph.babcanec@benedict.edu`

February 2, 2026

## Abstract

We present a new algorithm for computing arbitrary decimal digits of  $\pi$  without computing all preceding digits. Unlike Plouffe’s 2022 method based on Euler–Bernoulli number asymptotics, our approach derives directly from Machin’s classical arctangent formula. We introduce two techniques: series splitting, which transforms alternating arctangent series into non-alternating BBP-compatible components amenable to modular arithmetic, and a borrow-tracking lemma that enables combining fractional parts under subtraction without computing integer parts. The algorithm is implemented and verified for positions up to 10,000. This work demonstrates that classical Machin-type formulas, previously thought unsuitable for digit extraction, can be adapted for this purpose through systematic series decomposition.

**MSC 2020:** 11Y60, 11A63

**Keywords:** BBP formula, digit extraction, Machin’s formula

## 1 Introduction

The Bailey–Borwein–Plouffe formula, discovered in 1995 [1], fundamentally changed how mathematicians and computer scientists think about the computation of  $\pi$ . The formula,

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

possesses a remarkable property: it enables extraction of individual hexadecimal digits of  $\pi$  without computing any of the preceding digits. The formula’s power lies in its base-16 geometric factor, which allows modular arithmetic to extract fractional parts efficiently, yielding  $O(n \log^2 n)$  complexity for the  $n$ -th hexadecimal digit [2]. This represented a dramatic departure from all previous methods, which required computing the constant from the beginning to obtain any particular digit.

The situation for decimal digits has been considerably more challenging. Plouffe’s 1996 method [4] achieves  $O(n^3 \log^3 n)$  complexity using an identity involving central binomial coefficients, specifically

$$\sum_{n=1}^{\infty} \frac{2^n}{n \binom{2n}{n}} = \frac{\pi + 3}{2},$$

combined with fraction decomposition techniques. His more recent 2022 work [5] takes an entirely different approach, employing asymptotic formulas for Euler and Bernoulli numbers to obtain

$$\pi^n = \left( \frac{2(-1)^{n+1}(2n)!}{2^{2n}B_{2n}(1-2^{-n})(1-3^{-n})(1-5^{-n})(1-7^{-n})} \right)^{1/(2n)}$$

where  $B_{2n}$  denotes the Bernoulli numbers. This method enables extraction up to approximately the  $10^8$ -th decimal digit, though it remains fundamentally limited by the computational cost of evaluating large Euler numbers  $E_{2n}$  and does not constitute true digit extraction in the BBP sense, as it still requires substantial intermediate computation.

In this paper, we present an alternative approach based on Machin’s formula [6],

$$\pi = 16 \arctan \frac{1}{5} - 4 \arctan \frac{1}{239}, \quad (2)$$

which dates to 1706 and has been used extensively for high-precision computation of  $\pi$  due to its rapid convergence. Our key insight is that the factor  $1/5$  appearing in the first arctangent provides a natural base-5 structure. Combined with the elementary factorization  $10 = 2 \times 5$ , this observation enables decimal digit extraction through a hybrid base-2/base-5 approach that circumvents the apparent impossibility of finding a native base-10 BBP formula.

The main technical contributions of this work are as follows. First, we develop a series-splitting technique (Theorem 3.1) that decomposes the alternating arctangent series into non-alternating components suitable for BBP-style modular arithmetic. Second, we prove a borrow-tracking lemma (Lemma 5.1) showing that the fractional part  $\{A - B - C\}$  can be computed exactly from the individual fractional parts  $\{A\}$ ,  $\{B\}$ , and  $\{C\}$  alone, without any knowledge of the integer parts. Third, we provide a complete implementation that extracts decimal digits at arbitrary positions, verified against known values up to position 10,000.

The decomposition  $\arctan(1/5) = P_1/5 - Q_1/125$  into non-alternating BBP-compatible series appears to be new. While the borrow-tracking lemma is elementary, its application to digit extraction in this context is novel. Most significantly, this work provides the first demonstration that Machin-type formulas can be adapted for decimal digit extraction, contrary to the conventional wisdom that such formulas are unsuitable due to their alternating arctangent structure.

## 2 Preliminaries

We establish notation and recall the fundamental concepts underlying BBP-type digit extraction algorithms.

**Definition 2.1.** For  $x \in \mathbb{R}$ , the fractional part is defined as  $\{x\} = x - \lfloor x \rfloor$ , which lies in the interval  $[0, 1)$ .

**Definition 2.2.** The  $n$ -th decimal digit of  $\pi$  after the decimal point is given by

$$d_n = \lfloor 10 \cdot \{10^{n-1}\pi\} \rfloor.$$

**Definition 2.3** (BBP-Type Formula). A BBP-type formula has the general form

$$\alpha = \sum_{k=0}^{\infty} \frac{1}{b^k} \sum_{j=1}^m \frac{a_j}{(mk+j)^s}$$

where  $b \geq 2$  is the base, enabling extraction of base- $b$  digits via modular arithmetic [3].

The key property exploited in all BBP algorithms is that for integers  $a$  and  $d$  with  $d > 0$ , the fractional part satisfies

$$\left\{ \frac{b^n \cdot a}{d} \right\} = \frac{(b^n \cdot a) \bmod d}{d},$$

and crucially, the quantity  $(b^n \cdot a) \bmod d$  can be computed in  $O(\log n)$  operations via fast modular exponentiation without ever computing the potentially enormous full value  $b^n \cdot a$ . This observation is the foundation upon which all BBP-type extraction algorithms rest.

### 3 Series Splitting for Arctangent

The Taylor series for the arctangent function is given by

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

which converges for  $|x| \leq 1$ . The alternating signs present a fundamental obstacle to the direct application of modular arithmetic, since the BBP technique requires accumulating positive fractional contributions. We overcome this difficulty through a series-splitting approach that separates the positive and negative terms.

**Theorem 3.1** (Series Splitting). *Define the non-alternating series*

$$P_1 = \sum_{j=0}^{\infty} \frac{1}{(4j+1) \cdot 625^j}, \tag{3}$$

$$Q_1 = \sum_{j=0}^{\infty} \frac{1}{(4j+3) \cdot 625^j}. \tag{4}$$

Then

$$\arctan \frac{1}{5} = \frac{P_1}{5} - \frac{Q_1}{125}. \tag{5}$$

*Proof.* Starting from the Taylor series

$$\arctan \frac{1}{5} = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1) \cdot 5^{2k+1}},$$

we separate the sum according to the parity of the index. The even indices  $k = 2j$  contribute terms with coefficient  $+1$ , while the odd indices  $k = 2j+1$  contribute terms with coefficient  $-1$ . This yields

$$\begin{aligned} \arctan \frac{1}{5} &= \sum_{j=0}^{\infty} \frac{1}{(4j+1) \cdot 5^{4j+1}} - \sum_{j=0}^{\infty} \frac{1}{(4j+3) \cdot 5^{4j+3}} \\ &= \frac{1}{5} \sum_{j=0}^{\infty} \frac{1}{(4j+1) \cdot 625^j} - \frac{1}{125} \sum_{j=0}^{\infty} \frac{1}{(4j+3) \cdot 625^j} \\ &= \frac{P_1}{5} - \frac{Q_1}{125}, \end{aligned}$$

as claimed. □

**Remark 3.2.** The series  $P_1$  and  $Q_1$  are now BBP-compatible in the sense that they consist entirely of positive terms with geometric decay in base  $625 = 5^4$ . This structure enables the modular extraction of  $\{5^N \cdot P_1\}$  and  $\{5^N \cdot Q_1\}$  using the standard BBP technique.

Substituting the series splitting result into Machin's formula (2) and multiplying through by  $5^{N-1}$ , we obtain the following representation that forms the basis of our extraction algorithm.

**Corollary 3.3.** For any positive integer  $N$ ,

$$5^{N-1}\pi = 16 \cdot 5^{N-2}P_1 - 16 \cdot 5^{N-4}Q_1 - 4 \cdot 5^{N-1} \arctan \frac{1}{239}. \quad (6)$$

## 4 Modular Extraction of Components

We now show how to compute the fractional parts of the scaled series  $5^M \cdot P_1$  and  $5^M \cdot Q_1$  using modular arithmetic.

**Theorem 4.1** (Modular Extraction). For  $M \geq 0$ , the fractional part of  $5^M \cdot P_1$  is given by

$$\{5^M \cdot P_1\} = \left\{ \sum_{j=0}^{\lfloor M/4 \rfloor} \frac{5^{M-4j} \bmod (4j+1)}{4j+1} + \sum_{j > M/4} \frac{5^{M-4j}}{4j+1} \right\}. \quad (7)$$

The analogous formula holds for  $Q_1$  with denominators  $4j+3$  in place of  $4j+1$ .

*Proof.* Multiplying the definition of  $P_1$  by  $5^M$ , we have

$$5^M \cdot P_1 = \sum_{j=0}^{\infty} \frac{5^{M-4j}}{4j+1}.$$

For indices  $j$  satisfying  $j \leq M/4$ , the exponent  $M-4j$  is non-negative, so  $5^{M-4j}$  is a positive integer. We may therefore write each such term as

$$\frac{5^{M-4j}}{4j+1} = \left\lfloor \frac{5^{M-4j}}{4j+1} \right\rfloor + \frac{5^{M-4j} \bmod (4j+1)}{4j+1}.$$

The sum of the integer parts contributes only to  $\lfloor 5^M P_1 \rfloor$  and therefore vanishes when we take the fractional part of the total. For indices  $j > M/4$ , we have  $5^{M-4j} < 1$ , so these terms are already fractional and contribute directly to  $\{5^M \cdot P_1\}$  without requiring modular reduction.  $\square$

**Remark 4.2.** The modular exponentiation  $5^{M-4j} \bmod (4j+1)$  requires  $O(\log M)$  bit operations per term using the standard repeated-squaring algorithm. The tail sum over  $j > M/4$  converges geometrically with ratio  $1/625$ , so  $O(P)$  additional terms suffice to achieve  $P$  bits of precision.

## 5 The Borrow-Tracking Lemma

The expanded Machin formula (6) expresses  $5^{N-1}\pi$  as a difference of three terms. A naive approach to computing the fractional part of this difference would require tracking the integer parts of each term to handle potential borrows, which would negate the efficiency gains from modular extraction. The following lemma shows that this is unnecessary: the fractional part of a difference can be computed directly from the fractional parts of the individual terms.

**Lemma 5.1** (Borrow Tracking). *Let  $A, B, C \in \mathbb{R}$ . Define the borrow indicators  $\varepsilon_1, \varepsilon_2 \in \{0, 1\}$  by*

$$\varepsilon_1 = \begin{cases} 1 & \text{if } \{A\} < \{B\}, \\ 0 & \text{otherwise,} \end{cases} \quad (8)$$

$$\varepsilon_2 = \begin{cases} 1 & \text{if } \{A\} - \{B\} + \varepsilon_1 < \{C\}, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

*Then the fractional part of the difference  $A - B - C$  is given by*

$$\{A - B - C\} = (\{A\} - \{B\} - \{C\} + \varepsilon_1 + \varepsilon_2) \bmod 1. \quad (10)$$

*Proof.* Write  $A = I_A + F_A$  where  $I_A = \lfloor A \rfloor \in \mathbb{Z}$  and  $F_A = \{A\} \in [0, 1)$ , and decompose  $B$  and  $C$  similarly. For the first subtraction, we have

$$A - B = (I_A - I_B) + (F_A - F_B).$$

If  $F_A \geq F_B$ , then  $F_A - F_B \in [0, 1)$ , so  $\{A - B\} = F_A - F_B$  and no borrow occurs. If  $F_A < F_B$ , then  $F_A - F_B \in (-1, 0)$ , and we may rewrite

$$A - B = (I_A - I_B - 1) + (F_A - F_B + 1),$$

giving  $\{A - B\} = F_A - F_B + 1$ . In either case,  $\{A - B\} = F_A - F_B + \varepsilon_1$ .

Let  $F_{AB} = F_A - F_B + \varepsilon_1 = \{A - B\}$ , which lies in  $[0, 1)$ . Applying the same reasoning to the subtraction of  $C$ , we obtain  $\{(A - B) - C\} = F_{AB} - F_C + \varepsilon_2$  where  $\varepsilon_2 = 1$  if and only if  $F_{AB} < F_C$ .

Combining these results yields  $\{A - B - C\} = F_A - F_B - F_C + \varepsilon_1 + \varepsilon_2$ , reduced modulo 1 to lie in  $[0, 1)$ .  $\square$

**Remark 5.2.** *This lemma is the key to our method. It demonstrates that the subtraction in (6) can be performed using only the fractional parts of each component, with the borrow information determined entirely by simple comparisons between these fractional parts. No integer parts need be computed or stored at any point in the algorithm.*

## 6 Base Conversion and Digit Extraction

The series splitting and modular extraction techniques developed above allow us to compute  $\{5^{N-1}\pi\}$  efficiently. To extract decimal digits, we must convert from base 5 to base 10, which is accomplished by the following result.

**Theorem 6.1** (Base Conversion). *For any positive integer  $N$ ,*

$$\{10^{N-1}\pi\} = \{2^{N-1} \cdot \{5^{N-1}\pi\}\}. \quad (11)$$

*Proof.* Let  $5^{N-1}\pi = I + F$  where  $I = \lfloor 5^{N-1}\pi \rfloor$  is the integer part and  $F = \{5^{N-1}\pi\}$  is the fractional part. Then

$$10^{N-1}\pi = 2^{N-1} \cdot 5^{N-1}\pi = 2^{N-1}(I + F) = 2^{N-1}I + 2^{N-1}F.$$

Since  $I$  is an integer,  $2^{N-1}I$  is also an integer, and therefore  $\{10^{N-1}\pi\} = \{2^{N-1}F\}$ .  $\square$

**Corollary 6.2** (Digit Extraction). *The  $N$ -th decimal digit of  $\pi$  after the decimal point is*

$$d_N = \lfloor 10 \cdot \{2^{N-1} \cdot \{5^{N-1}\pi\}\} \rfloor.$$

## 7 Algorithm

Algorithm 1 presents the complete procedure for extracting the  $N$ -th decimal digit of  $\pi$ .

---

**Algorithm 1** Decimal Digit Extraction for  $\pi$

---

**Require:**  $N \geq 1$

**Ensure:**  $d_N$ , the  $N$ -th decimal digit of  $\pi$  after the decimal point

```

1: Set precision  $P \leftarrow 2N + 100$  bits

2: // Compute fractional parts using modular arithmetic
3:  $F_P \leftarrow \{5^{N-2} \cdot P_1\}$  ▷ Theorem 4.1
4:  $F_Q \leftarrow \{5^{N-4} \cdot Q_1\}$ 
5:  $F_T \leftarrow \{5^{N-1} \cdot \arctan(1/239)\}$  ▷ Direct computation;  $O(N)$  terms

6: // Apply scaling factors and reduce mod 1
7:  $F_A \leftarrow \{16 \cdot F_P\}$ 
8:  $F_B \leftarrow \{16 \cdot F_Q\}$ 
9:  $F_C \leftarrow \{4 \cdot F_T\}$ 

10: // Combine using borrow tracking (Lemma 5.1)
11:  $\varepsilon_1 \leftarrow [F_A < F_B]$  ▷ Iverson bracket: 1 if true, 0 if false
12:  $F_{AB} \leftarrow F_A - F_B + \varepsilon_1$ 
13:  $\varepsilon_2 \leftarrow [F_{AB} < F_C]$ 
14:  $F_{5\pi} \leftarrow (F_{AB} - F_C + \varepsilon_2) \bmod 1$ 

15: // Convert to base 10 and extract digit
16:  $F_{10\pi} \leftarrow \{2^{N-1} \cdot F_{5\pi}\}$ 
17: return  $\lfloor 10 \cdot F_{10\pi} \rfloor$ 

```

---

## 8 Implementation and Verification

The algorithm has been implemented in Python using arbitrary-precision decimal arithmetic provided by the standard library `decimal` module. Table 1 presents computed digits at selected positions, all of which have been verified against published digit tables [8]. The implementation requires no external dependencies or precomputed tables of any kind.

## 9 Comparison with Prior Work

The original BBP formula [1] achieves  $O(n \log^2 n)$  complexity for hexadecimal digits because the base 16 allows direct digit extraction without any base conversion step, each term requires only  $O(\log n)$ -bit modular arithmetic, and the sum of  $O(n)$  fractional parts in  $[0, 1)$  needs only  $O(\log n)$  precision per term. Our method necessarily differs in requiring a base-5 to base-10 conversion via Theorem 6.1, which introduces additional precision requirements in the final multiplication by  $2^{N-1}$ .

Position	Digit	Time (s)
100	9	0.001
500	2	0.02
1,000	9	0.11
2,000	9	0.72
5,000	1	9.9
10,000	8	61.6

Table 1: Computed decimal digits of  $\pi$ . Position uses 1-indexing after the decimal point, so position 1 corresponds to the digit 1 in  $\pi = 3.14159\dots$ . All results verified against [8].

Plouffe’s 1996 method [4] takes an entirely different approach, exploiting the identity involving central binomial coefficients mentioned in the introduction and achieving  $O(n^3 \log^3 n)$  complexity through fraction decomposition techniques. His 2022 method [5] based on Bernoulli and Euler number asymptotics represents a significant theoretical advance but remains limited in practice by the computational cost of evaluating large special numbers, with practical extraction limited to approximately  $n \leq 10^8$  using current methods for computing Bernoulli numbers.

Our approach is complementary to these existing methods. It uses only classical series derived from Machin’s formula without requiring any precomputed special number sequences, and it introduces new techniques—series splitting and borrow tracking—that may find applications to other mathematical constants expressible in terms of arctangent identities. While not achieving the optimal  $O(n \log^2 n)$  complexity of hexadecimal BBP, the method demonstrates that Machin-type formulas, which have been used for centuries to compute  $\pi$  to high precision, can also serve as a foundation for digit extraction algorithms.

## 10 Conclusion

We have presented a new method for extracting decimal digits of  $\pi$  based on Machin’s formula. The key innovations are series splitting, which eliminates the alternating signs that would otherwise prevent the application of modular arithmetic, and borrow tracking, which enables the combination of fractional parts under subtraction without computing or storing any integer parts.

The algorithm has been implemented and verified for positions up to 10,000. While not achieving the  $O(n \log^2 n)$  complexity of hexadecimal BBP due to the base conversion step, the method establishes that Machin-type formulas can be adapted for decimal digit extraction, contrary to the conventional wisdom that such formulas are unsuitable for this purpose due to their alternating arctangent structure. This opens the possibility of applying similar techniques to other mathematical constants that admit representations in terms of arctangent series.

## References

- [1] D. H. Bailey, P. Borwein, and S. Plouffe, On the rapid computation of various polylogarithmic constants, *Math. Comp.* **66** (1997), 903–913.

- [2] D. H. Bailey, The BBP algorithm for pi, Technical report, Lawrence Berkeley National Laboratory, 2006.
- [3] D. H. Bailey, A compendium of BBP-type formulas for mathematical constants, Technical report, 2000.
- [4] S. Plouffe, On the computation of the  $n$ 'th decimal digit of various transcendental numbers, Unpublished manuscript, November 1996.
- [5] S. Plouffe, A formula for the  $n$ th decimal digit or binary of  $\pi$  and powers of  $\pi$ , arXiv:2201.12601, 2022.
- [6] J. Machin, The ratio of the radius to the circumference, *Philos. Trans. Roy. Soc. London*, 1706.
- [7] F. Bellard, A new formula to compute the  $n$ -th binary digit of pi, Technical report, 1997.
- [8] Various authors, Tables of digits of  $\pi$ , <https://www.angio.net/pi/digits.html>