

圆周率近似值的计算

庄裕旻

2024 年 1 月 10 日

目录

一、引言	1
二、有关圆周率近似值计算的数学史	1
三、几个重要的可用于计算圆周率的方法的证明	3
1. 阿基米德迭代法的证明	3
2. 格雷戈里-莱布尼茨级数的证明	4
3. 牛顿的级数展开式的证明	4
4. 拉马努金公式的证明（简述）	5
四、圆周率的计算实践	6
1. 基于阿基米德迭代法的计算程序	6
2. 基于拉马努金公式的计算程序	7
3. 基于 Gauss-Salamin-Brent 迭代法的计算程序	7
五、总结	9

摘要

π 是一个重要的常数，自从人类认识圆以来，便一直在优化计算 π 的方法，本文将简要梳理有关圆周率的数学史，总结一些重要的迭代法或级数的证明，并选取其中效率较高的阿基米德迭代法、拉马努金公式和 Gauss-Salamin-Brent 迭代法进行实际计算测试。

关键词：圆周率计算，阿基米德迭代法，拉马努金公式，Gauss-Salamin-Brent 迭代法

一、引言

圆周率是数学中的一个重要常数，它可以算是数学常数中与现实世界的关联最为密切的常数之一，无论是在基础科学研究，还是在工程技术应用中，都有其广泛而重要的作用。然而，尽管圆周率的概念非常简单，即圆的周长与直径的比值，但其精确值却无法用有限的数字来表示，这就引出了一个有趣而重要的问题：如何计算出圆周率的近似值，以满足各种实际需要？这是一个自古至今，从东方到西方，数学家们一直在探索和研究的课题。

本文将对圆周率近似值的计算方法进行深入研究，首先，本文将回顾有关圆周率近似值计算的数学史，展现人类对这个问题的探索过程和成果；然后，本文将介绍一些重要的计算圆周率近似值的方法，并对其进行证明；最后，本文将选取其中几种效率较高的方法，进行实际计算测试，验证其有效性和准确性。希望通过本文的总结与分析，能够对圆周率近似值的计算有更深入的理解。

二、有关圆周率近似值计算的数学史

圆周率 π 的发现与计算过程是一个漫长的过程^[1]，从一开始发现圆的周长和直径的比值是一个常数，到开始基于圆的几何性质进行估算，再到数列和级数发明后用全新的方式进行估算，圆周率近似值的计算过程是几何学和分析学发展的一个缩影。

在认识圆周率的早期，主要的发展成就是“圆和圆周率是什么”。最早的圆周率记载，大约是在 4000 多年前的古巴比伦文明^[1][Page 21]，从目前出土的石刻来看，古巴比伦人，将圆周率确定在了 3.125。而在 2500 年前的古代埃及，虽然没有确凿的证据，但是胡夫金字塔的周长和高度之比刚好是圆周率的两倍左右，甚至据说准确度已达到后世割圆术的精度（但考虑到后人测量本身的误差，笔者对这个结论持怀疑态度），所以我们可以猜测，古埃及金字塔的建筑，和圆周率是存在关系的。而中国有关圆周率的最早记载，是春秋战国时期的墨家鲁班书籍当中，出现了“圆径一而周三”的说法。公元前五世纪，伊里斯的希比亚斯^[1][Page 40] 试图通过希比亚斯四等分线来确定圆周率。

随着几何学的进一步发展，数学家们开始尝试通过几何方法来估算圆周率的值^[1]。公元前三世纪，阿基米德通过圆内接正多边形和外接正多边形来逼近圆，并由此计算得 $3\frac{10}{71} < \pi < 3\frac{1}{7}$ 。公元 5 世纪，在刘徽“割圆术”的基础上，祖冲之计算得 $3.1415926 < \pi < 3.1415927$ ，虽然祖冲之所著《缀术》失传，但有人证明了通过算筹方便地进行高精度开方运算是可行的^[2]，并且从名称上来看很可能与真正的《缀术》确实有关。16 世纪的列奥纳多·达·芬奇和 17 世纪时的一份日本文献记载了一种通过重新排列求解圆的近似面积的方式^[1][Page 19]，这种方法将圆分成相等的扇形，并将它们排列成一个近似的平行四边形。这种方法并不复杂，因此有人怀疑古代人们可能使用了这种方法的一种形式来计算圆的面积。

到了 17 和 18 世纪，随着数列和级数的发现，圆周率近似值的计算进入全新时期^[1]。格雷戈里、莱布尼茨和欧拉等数学家发现了 π 的无穷级数表示，使得可以相对容易地计算 π 到小数点后几十位。不久后牛顿优化了 π 的级数求解方式，极大地加快了收敛速度，仅仅使用 22 项，就能得到小数点后 16 位的精度。在接下来的一两百年里，众多数学家，如傅里叶、黎曼等，提出了许多收敛于 π 的级数^[3]，但是收敛速度从主观上来讲暂无明显提升。

20 世纪初，拉马努金公式的发现使得收敛速度明显加快，电子计算机的发明结合拉马努金公式等收敛速度较快的计算方式，使得已知的圆周率位数飞速上升，1949 年时，通过计算机 ENIAC，人类将 π 估算至小数点后 2035 位，在 1967 年，通过计算机 CDC⁶⁶⁰⁰，将 π 计算至小数点后 500000 位^[4]。在 1976 年的两篇论文中，Salamin^[5]和 Brent^[6]基于算术几何平均和高斯的想法，独立给出了一个比其他任何圆周率经典公式收敛快得多的迭代算法，一种称法是 Gauss-Salamin-Brent 迭代算法，它可以二次收敛于 π ，本质性地提升了收敛速度，开启了圆周率近似值计算的新纪元。很快，在 1987 年，圆周率的估算精度就达到了亿位级别^[4]。

从最初连概念都是模糊的，到如今有效数字突破万亿级，圆周率近似值计算的发展，是人类数学发展的一个缩影，它代表了人类对数学的认知，从感性走向理性，又用理性成就实用的历史演进过程。

三、几个重要的可用于计算圆周率的方法的证明

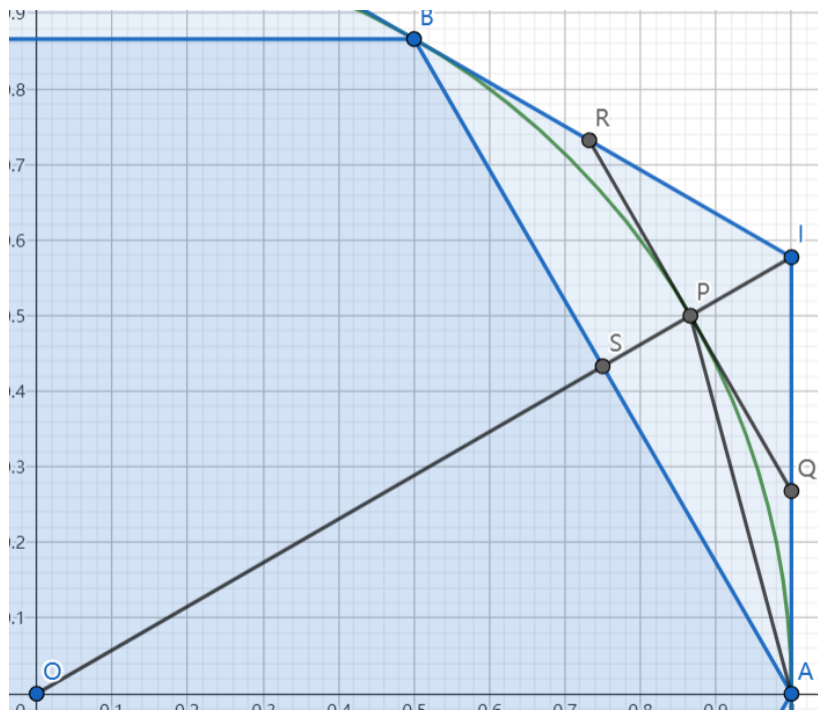
1. 阿基米德迭代法的证明

阿基米德迭代法^[7]的表达式如下：

$$a_0 = 2\sqrt{3}, b_0 = 3$$

$$a_k = \frac{2a_{k-1}b_{k-1}}{a_{k-1} + b_{k-1}}, b_k = \sqrt{a_k b_{k-1}}$$

$$\lim_{k \rightarrow +\infty} a_k = \lim_{k \rightarrow +\infty} b_k = \pi$$



证明. 取一个半径为 1 的圆，则它的半周长就是 π ，设 a_i 为外切正多边形的半周长， b_i 为内接正多边形的半周长，初始时均取六边，则有 $a_0 = 2\sqrt{3}, b_0 = 3$ 。不失一般性，假设内外两个正多边形的切点和接点重合，且每次“割圆”后边数翻倍。

先证明 a_k 的递推式：

$$\frac{a_k}{a_{k-1}} = \frac{PQ + AQ}{IQ + AQ} = \frac{2PQ}{IQ + PQ}$$

由于 $\triangle IPQ \sim \triangle ISA$ ，有

$$\frac{a_k}{a_{k-1}} = \frac{PQ + AQ}{IQ + AQ} = \frac{2PQ}{IQ + PQ} = \frac{2SA}{IA + SA} = \frac{2b_{k-1}}{a_{k-1} + b_{k-1}}$$

即可证明

$$a_k = \frac{2a_{k-1}b_{k-1}}{a_{k-1} + b_{k-1}}$$

然后证明 b_k 的递推式：

$$\frac{b_k}{b_{k-1}} = \frac{2AP}{AB} = \frac{AP}{AS}$$

由几何关系和三角函数易得 $\frac{PQ+AQ}{AS} = \left(\frac{AP}{AS}\right)^2$, 由此可推出:

$$\frac{b_k}{b_{k-1}} = \sqrt{\frac{a_k}{b_{k-1}}}, b_k = \sqrt{a_k b_{k-1}}$$

当 $k \rightarrow +\infty$ 时, 内接正多边形和外切正多边形均“趋向于”和圆重合, 此时 $a_k \rightarrow \pi, b_k \rightarrow \pi$ 。
证毕

2. 格雷戈里-莱布尼茨级数的证明

格雷戈里-莱布尼茨级数由莱布尼兹基于格雷戈里的反正切函数展开式提出, 该级数的表达式如下:

$$\frac{\pi}{4} = \sum_{n=0}^{+\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

证明. 首先, 求解反正切函数的级数展开式。

由于

$$(\arctan x)' = \frac{1}{1+x^2}$$

因此

$$\arctan(x) = \int_0^x \frac{1}{1+t^2} dt$$

对 $\frac{1}{1+t^2}$ 进行几何级数展开, 得到:

$$\frac{1}{1+t^2} = \sum_{n=0}^{+\infty} (-t^2)^n \quad (|t| < 1)$$

则有

$$\arctan(x) = \int_0^x \frac{1}{1+t^2} dt = \int_0^x \sum_{n=0}^{+\infty} (-t^2)^n dt = \sum_{n=0}^{+\infty} (-1)^n \int_0^x t^{2n} dt = \sum_{n=0}^{+\infty} \frac{(-1)^n x^{2n+1}}{2n+1} \quad (|x| < 1)$$

接下来, 代入 $x = 1$, 则有:

$$\frac{\pi}{4} = \arctan(1) = \sum_{n=0}^{+\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

证毕

3. 牛顿的级数展开式的证明

牛顿的级数展开式基于反正弦函数的展开式, 该级数的表达式如下:

$$\frac{\pi}{6} = \sum_{n=0}^{+\infty} \frac{(2n+1)!!}{(2n)!!(2n+1)^2 2^{2n+1}}$$

证明. 先证明反正弦函数的级数展开式。这一步可以用仿照格雷戈里-莱布尼茨级数的证明思路，先把 $\arcsin(z)$ 写成 $\int_0^z \frac{1}{\sqrt{1-t^2}} dt$ ，然后将 $\frac{1}{\sqrt{1-t^2}}$ 表示成幂级数展开，通过二项式定理，得到：

$$\frac{1}{\sqrt{1-t^2}} = (1-t^2)^{-\frac{1}{2}} = \sum_{n=0}^{+\infty} \binom{-\frac{1}{2}}{n} (-t^2)^n = \sum_{n=0}^{+\infty} \frac{C_{2n}^n}{2^{2n}} t^{2n}$$

则有：

$$\arcsin(y) = \sum_{n=0}^{+\infty} \frac{C_{2n}^n}{2^{2n}} \frac{y^{2n+1}}{2n+1}$$

代入 $\frac{1}{2}$ ，有：

$$\frac{\pi}{6} = \arcsin\left(\frac{1}{2}\right) = \sum_{n=0}^{+\infty} \frac{(2n+1)!!}{(2n)!!(2n+1)^2 2^{2n+1}}$$

■

4. 拉马努金公式的证明（简述）

以这个版本为例（主要借鉴自 *On proving some of Ramanujan's formulas for $\frac{1}{\pi}$ with an elementary method*^[8]）：

$$\frac{9}{2\pi} = \sum_{n=0}^{+\infty} \frac{\left(\frac{1}{2}\right)_n \left(\frac{1}{4}\right)_n \left(\frac{3}{4}\right)_n}{(1)_n^3} (7n+1) \left(\frac{32}{81}\right)^n$$

这个“elementary”的证明方法，主要基于转换法。该论文首先梳理了拉马努金早年的一些文献，并结合其他已有结论，确定了若干“出发点”；随后，作者总结了所有已知的拉马努金公式，归纳出了一个普遍的形式，并在此基础上进行了一些化归和整理，例如该版本归属于 $s = \frac{1}{4}$ 这一类；接着，作者完善了“出发点”和相关引理的证明，使得后续可以基于它们进行转化；然后向已知结论化归^[8][Page 26]：

$${}_3F_2\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1, x\right) = (1-x)^{-\frac{1}{2}} {}_3F_2\left(\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, 1, 1, -\frac{4x}{(1-x)^2}\right)$$

四、圆周率的计算实践

基于前述的理论推导，选取效率较高的三种计算方法进行实际计算检验，计算过程使用 Python 的 mpmath 提供高精度浮点数运算，并在完成计算后与标准答案^[9]进行比对，当程序输出的 π 近似值在小数点后 10000 位内与标准答案相同，考虑到 π 的十进制表示难以出现“末位的微小误差引起大量数位的改变”这种情况，可以认为该程序计算结果的前 10000 位是准确的。完整的程序源码、运行环境及数据等信息详见<https://dev.zymsite.ink/PiCalculator>。

1. 基于阿基米德迭代法的计算程序

基于阿基米德迭代法^[7]

$$a_0 = 2\sqrt{3}, b_0 = 3$$

$$a_k = \frac{2a_{k-1}b_{k-1}}{a_{k-1} + b_{k-1}}, b_k = \sqrt{a_k b_{k-1}}$$

$$\lim_{k \rightarrow +\infty} a_k = \lim_{k \rightarrow +\infty} b_k = \pi$$

可以写出如下程序：

```
#!/usr/bin/pypy3
from mpmath import mp

# 设置精度（位数）
mp.dps = 10000

a_0=mp.mpf(2)*mp.sqrt(3)
b_0=mp.mpf(3)
for k in range(20000):
    a_k=2*a_0*b_0/(a_0+b_0)
    b_k=mp.sqrt(a_k*b_0)
    a_0=a_k
    b_0=b_k

approx_pi=(a_0+b_0)/2
print(approx_pi)
```

虽然阿基米德迭代法中，每一步计算的运算量较小，但由于它的迭代速度相对较慢，每一步迭代只能增加 0 到 2 位有效数字^[7]，因此在这种具体实现下（中间过程精度受限，也是 10000 位），精确到小数点后 10000 位需要迭代约 20000 步。

2. 基于拉马努金公式的计算程序

通过拉马努金公式中，这个最有名的版本^[10]

$$\frac{1}{\pi} = \frac{\sqrt{8}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!}{(n!)^4} \cdot \frac{1103 + 26390n}{396^{4n}}$$

可以写出如下程序：

```
#!/usr/bin/pypy3
from mpmath import mp

# 设置精度（位数）
mp.dps = 10000

def compute_pi(n_terms: int) -> mp.mpf:
    pi_inverse = mp.mpf(0)
    for k in range(n_terms):
        numerator = mp.fac(4*k) * (1103 + 26390*k)
        denominator = mp.power(mp.fac(k), 4) * mp.power(396, 4*k)
        term = (2 * mp.sqrt(2) / 9801) * (numerator / denominator)
        pi_inverse += term
    return 1 / pi_inverse

# 计算的近似值，使用2000个项
approx_pi = compute_pi(2000)
print(approx_pi)
```

虽然这个公式每增加一项可以增加约 8 位有效数字^[7]，但由于在这种具体实现下，运算过程中小数点后的数字就已经被截断，因此达到 10000 位有效数字需要大约计算 2000 项。

3. 基于 Gauss-Salamin-Brent 迭代法的计算程序

基于 Gauss-Salamin-Brent 迭代公式^[7]

$$a_0 = 1, b_0 = \frac{\sqrt{2}}{2}, s_0 = \frac{1}{2}$$

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

$$\lim_{k \rightarrow +\infty} p_k = \pi$$

可以写出如下程序

```
#!/usr/bin/pypy3
from mpmath import mp

# 设置精度（位数）
mp.dps = 11000

a_0=mp.mpf(1)
```



```
b_0=mp.sqrt(2)/mp.mpf(2)
s_0=mp.mpf(0.5)
for k in range(1,13+1):
    a_k=(a_0+b_0)/2
    b_k=mp.sqrt(a_0*b_0)
    s_k=s_0-mp.power(2,k)*(a_k*a_k-b_k*b_k)
    a_0=a_k
    b_0=b_k
    s_0=s_k
approx_pi=2*a_0*a_0/s_0
print(approx_pi)
```

由于涉及到 $a_k^2 - b_k^2$ 这样的项，所以对中间过程的保留精度要求略高一些，中间过程的保留位数要超过 10000 位才有可能最终结果精确到 10000 位，在这种具体实现方式下，精确到 10000 位需要 13 次迭代

以上的三种方法中，阿基米德迭代法本质上基于割圆术一类的思想，最早的达到实用精度的 π 近似值便是采用类似的方式手工计算出的^[11]；基于拉马努金公式的计算方法使得收敛速度明显加快，是电子计算机时代圆周率近似值计算的重要里程碑；而 Gauss-Salamin-Brent 迭代法为二次收敛^[7]，可以在对数步迭代内完成近似计算，计算效率远超旧有的计算方法，是圆周率计算领域的飞跃性进步。

五、总结

本文通过对圆周率近似值的计算方法的简单介绍，梳理了人类对这一重要数学常数的认识和计算方法的演进过程。从最早的几何方法，到后来的线性收敛的级数和迭代等方法，再到现代的非线性收敛的方法，每一种方法的出现都代表了数学理论和计算技术的重大进步。

通过对阿基米德迭代法、拉马努金公式和 Gauss-Salamin-Brent 迭代法等方法进行理论证明和实际计算测试，本文验证了这些方法的有效性和准确性。特别是 Gauss-Salamin-Brent 迭代法，其二次收敛的特性使得计算效率远超旧有的计算方法，为圆周率计算领域带来了飞跃性的进步。

然而，尽管我们已经可以计算出圆周率的近似值到小数点后万亿位，但这并不意味着我们已经完全掌握了圆周率。圆周率的无理性和超越性仍然使其充满了神秘和魅力，如何更高效地计算圆周率，如何更深入地理解圆周率，以及圆周率与其他数学领域的关系等问题，仍然需要我们去探索和研究。事实上，在 1991 年，Jonathan Borwein 和 Peter Borwein 提出了一种三次收敛的算法^[12]，进一步提升了计算效率。数学是永无止境的，这是数学的魅力所在，它总是能提供无穷无尽的问题和挑战，激发我们的好奇心和求知欲。

References

- [1] BECKMANN P. A History of [Pi] (Pi)[M/OL]. In collab. with Internet Archive. New York : St. Martin's Press, 1971 [2024-01-10]. http://archive.org/details/historyofpypi0000beck_g8t1. 214 pp. ISBN: 978-0-312-38185-1 (引用页: 1, 2).
- [2] 石像无言. 祖冲之是怎么计算圆周率的? [EB/OL]. 知乎专栏. (2023-11-05) [2024-01-10]. <https://zhuanlan.zhihu.com/p/445101862> (引用页: 1).
- [3] DreamSnake. 圆周率 pi 比较著名的无穷级数公式有哪些? - 知乎[EB/OL]. (2020-07-19) [2024-01-09]. <https://www.zhihu.com/question/402311979/answer/1339659450> (引用页: 2).
- [4] 孙宏安. 圆周率计算简史[J/OL]. 中学数学教学参考, 1998(11): 49-50 [2024-01-09]. https://kns.cnki.net/kcms2/article/abstract?v=vdPasdvfHvu1_KpCSFzSEIx5Qp3GuQwWTfAjedXyDKyCA2DTRtW849Zbw8ENmNR9nwUv-F-Il4QukPQYl3vmRBIFnXOWn_j97H4Wh90FfEkHA3kSEtDCyhJqTJ3ME-eFL1GP5gsSpbI=&uniplatform=NZKPT&language=CHS. ISSN: 1002-2172 (引用页: 2).
- [5] SALAMIN E. Computation of π Using Arithmetic-Geometric Mean (1976)[G/OL]//BAILEY D H, BORWEIN J M. Pi: The Next Generation: A Sourcebook on the Recent History of Pi and Its Computation. Cham: Springer International Publishing, 2016: 1-8 [2024-01-10]. https://doi.org/10.1007/978-3-319-32377-0_1. DOI: 10.1007/978-3-319-32377-0_1. ISBN: 978-3-319-32377-0 (引用页: 2).
- [6] BRENT R P. Fast Multiple-Precision Evaluation of Elementary Functions[J/OL]. Journal of the ACM, 1976, 23(2): 242-251(1976-04-01) [2024-01-10]. <https://dl.acm.org/doi/10.1145/321941.321944>. DOI: 10.1145/321941.321944. ISSN: 0004-5411 (引用页: 2).
- [7] 石勇国, 宋涓, 王巧玲, 等. 计算圆周率方法的数值实验比较[J/OL]. 高等数学研究, 2022, 25(4): 24-27 [2024-01-09]. https://kns.cnki.net/kcms2/article/abstract?v=vdPasdvfHvtTpM6qAXiSn3gOmyNHLs4OrgUJyCbdt74YnckFB3tz6ww_vfKUU5c-pYBe1sbgf40K0I-vth_yUZV3PrIt4ixJIAGk3jH2Xgqk_1UBcTWptRCB71IEtJURR9mpN-YiamnVbOufYFZHEA==&uniplatform=NZKPT&language=CHS. ISSN: 1008-1399 (引用页: 3, 6-8).
- [8] AYCOCK A. On Proving Some of Ramanujan's Formulas for $\frac{1}{\pi}$ with an Elementary Method[EB/OL]. (2013-09-04) [2024-01-09]. <http://arxiv.org/abs/1309.1140>. arXiv: 1309.1140 [math]. preprint (引用页: 5).
- [9] Michael D. Huberty, Ko Hayashi, Chia Vang. 100,000 Digits of Pi[EB/OL]. (1997-07-06) [2024-01-09]. <http://www.geom.uiuc.edu/~huberty/math5337/groupe/digits.html> (引用页: 6).

- [10] BORWEIN J M, BORWEIN P B, BAILEY D H. Ramanujan, Modular Equations, and Approximations to Pi or How to Compute One Billion Digits of Pi[J/OL]. The American Mathematical Monthly, 1989, 96(3): 201-219. JSTOR: 2325206 [2024-01-09]. <https://www.jstor.org/stable/2325206>. DOI: 10.2307/2325206. ISSN: 0002-9890 (引用页: 7).
- [11] Michael D. Huberty, Ko Hayashi, Chia Vang. Historical Overview of Pi[EB/OL]. (1997-07-06) [2024-01-09]. <http://www.geom.uiuc.edu/~huberty/math5337/groupe/overview.html> (引用页: 8).
- [12] Borwein's Algorithm. //Wikipedia. 2022 [2024-01-10]. https://en.wikipedia.org/w/index.php?title=Borwein%27s_algorithm&oldid=1115360750 (引用页: 9).
- [13] Ben Lynn. Pi - Ramanujan's Formula for Pi[EB/OL]. 2000 [2024-01-09]. <https://crypto.stanford.edu/pbc/notes/pi/ramanujan.html>.
- [14] 甘泉. 关于圆周率 pi 的数值计算的一个改进公式[J/OL]. 中国科技信息, 2011(10): 45 [2024-01-09]. <https://kns.cnki.net/kcms2/article/abstract?v=vdPasdvfHvtDa2Di7EelCWVwubNX2BWLCCuEdRmKRxpmsEiesgL7zb9BNTcIOEC2ivCTEdiI0aVUnBwUFx8lg1twdxrgXwWBNdQZWQoVrIwwwNv3efj-JQ5fsfmd5eCZ7kZyhjobkjs=&uniplatform=NZKPT&language=CHS>. ISSN: 1001-8972.
- [15] GUILLERA J. A Method for Proving Ramanujan Series for $1/\pi$ [M]. 2018.
- [16] 罗洪斌, 钱伟, 马梅玲. 圆周率的计算方法及其 MATLAB 编程实现[J/OL]. 数字技术与应用, 2016(8): 57-58 [2024-01-09]. https://kns.cnki.net/kcms2/article/abstract?v=vdPasdvfHvuQ9ErlaH6iszFZaDG74Qt54so3D4SHoayJSb0y8Xc6E5C6nO2yjf5bB4mJdKsT6mwD83jTGf05oEskaHnKu6U-VHa58pwK1s5hYfJ-5af4k41rnpbANRBSr3LgLawEH5fgi7ECy_4zAA==&uniplatform=NZKPT&language=CHS. DOI: 10.19695/j.cnki.cn12-1369.2016.08.045. ISSN: 1007-9416.
- [17] Ron Knott. Pi and the Fibonacci Numbers[EB/OL]. (2023-08-30) [2024-01-10]. <https://r-knott.surrey.ac.uk/Fibonacci/fibpi.html#section1.7>.
- [18] SALAMIN E. Computation of π Using Arithmetic-Geometric Mean[J/OL]. Mathematics of Computation, 1976, 30(135): 565-570. JSTOR: 2005327 [2024-01-10]. <https://www.jstor.org/stable/2005327>. DOI: 10.2307/2005327. ISSN: 0025-5718.
- [19] 涂泓, 冯承天. 关于圆周率 π 的十个表达式[J/OL]. 上海中学数学, 2018(9): 18-19+37 [2024-01-09]. https://kns.cnki.net/kcms2/article/abstract?v=vdPasdvfHvuOoruVTFTVh9oJDk4h0qaZiCHj7vdA-eXpNeATNwwJwNQITyx31CbRflA09Y-YJsWWGfqtoeIlFBQicBTG1ak5BmyzQgZmH6C1_gonU3Sz7x5xxnrogLlg7Pi5IrVKlF7qG2eorKsg==&uniplatform=NZKPT&language=CHS. ISSN: 1672-7495.