B1 Numerical Algorithms: Computing value of π

Jin Rhee

4 November 2024

Introduction This report discusses three different methods to estimate the value of π and details of their respective implementation and error. Code was developed in C++17 on Apple M1 arm64. Code: `https://github.com/JinRhee/b1-numerical-algorithms-practical`

Method and Implementation Three methods are used to calculate an estimate of π:

Monte Carlo. This method is implemented by picking N pairs (x, y) of random numbers (from independent uniform real distributions), and returning the ratio $\frac{N'}{N} \times 4$ where N' is the number of pairs that satisfy $x^2 + y^2 \leq 1$. Multiple number of samples are tested $N = 2^k,\ k \in [0, 25]$ with relative error and error bar considerations.

Newton-Raphson. This root-finding algorithm is applied to $f(x) = \sin(x)$ which has the roots $x = n\pi,\ n \in \mathbb{Z}$. The implementation iterates $x_n = x_{n-1} + \tan x_{n-1}$ and terminates when $|x_n - x_{n-1}| \leq \epsilon$ (i.e. at convergence criteria), returning x_n. Multiple error values are tested (default $\epsilon = 0.1$) and the algorithm is given an initial estimate of $x_0 = 2$.

Chudnovsky algorithm. This algorithm 1 is proposed as the third method to estimate π. In short, this algorithm calculates π as the sum of an infinite series.[2]

Error All three methods are implemented in 8-byte `double` which offers a maximum precision to 16 decimal points.

Monte Carlo. The Central Limit Theorem states that the sample average approaches the normal distribution as the sample size increases. Due to this, we can use the sample variance to plot error bars representing one standard deviation.1 Further, the relative error of the estimate is plotted against the sample size, showing its proportionality to $\frac{1}{\sqrt{N}}$.[3]

Newton-Raphson. An analysis using the Taylor expansion shows that the error propogates quadratically $\epsilon_{n+1} \propto \epsilon_n^2$ at each iteration.[1] The quadratic error propogation can be used with the initial error to quantify error at some iteration: $\epsilon_n \propto \epsilon_0^{2^n}$

Chudnovsky algorithm. Error originates from the truncation of the infinite series and the precision of the floating point representation of the `double` data type. The later is dominant, as empirically the series sum of more than two terms returns the same value.

Discussion The Chudnovsky algorithm is most efficient and accurate method out of the three. Though requiring (costly) factorial computation, it needs only two terms of the series to achieve a higher accuracy than that of Newton-Raphson or Monte Carlo within the 16 decimal limit set by the `double` data type.

Empirically, [`Chud., Newt., Mont.`] take [`0.00025ms, 0.00063ms, 1551.81ms`] to produce absolute errors of [`2.93-14, 5.00e-08, 3.92e-04`]. Due to both superior accuracy and efficiency on the tested implementation, the Chudnovsky algorithm is recommended for estimating π.

Appendix

Figures

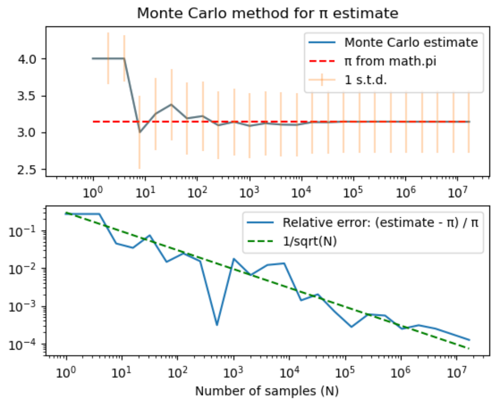

Figure 1: Monte Carlo method error plots.

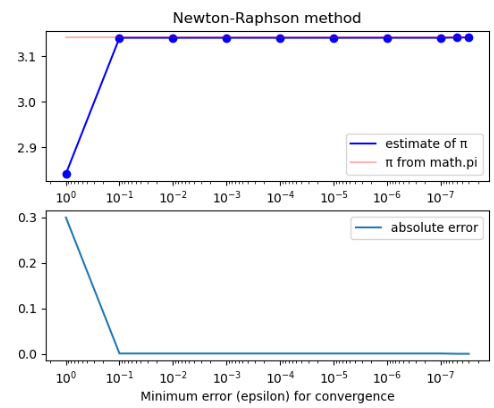

Figure 2: Newton-Raphson error plots.

Equations

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

References

[1] Joel Feldman. *Error Behaviour of Newton's Method*. Accessed: 3-November-2024. URL: `https://personal.math.ubc.ca/~feldman/m120/newtConv.pdf`.

[2] Wikipedia contributors. *Chudnovsky algorithm — Wikipedia, The Free Encyclopedia*. [Online; accessed 3-November-2024]. 2024. URL: `https://en.wikipedia.org/w/index.php?title=Chudnovsky_algorithm&oldid=1253211016`.

[3] Wikipedia contributors. *Monte Carlo integration — Wikipedia, The Free Encyclopedia.* [Online; accessed 3-November-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=Monte_Carlo_integration&oldid=1255738066.