Borwein's algorithm

From Wikipedia, the free encyclopedia

In mathematics, **Borwein's algorithm** is an algorithm devised by Jonathan and Peter Borwein to calculate the value of 1/π. They devised several other algorithms. They published a book: Jonathon M. Borwein, Peter B. Borwein, Pi and the AGM - A Study in Analytic Number Theory and Computational Complexity, Wiley, New York, 1987. Many of their results are available in: Jorg Arndt, Christoph Haenel, Pi Unleashed, Springer, Berlin, 2001, ISBN 3-540-66572-2.

Contents [hide]

Jonathan Borwein and Peter Borwein's Version (1993) [edit]

Start out by setting[citation needed]

$$\begin{aligned}
A &= 63365028312971999585426220 \\
&\quad + 28337702140800842046825600\sqrt{5} \\
&\quad + 384\sqrt{5}(10891728551171178200467436212395209160385656017 \\
&\qquad + 4870929086578810225077338534541688721351255040\sqrt{5})^{1/2} \\
B &= 7849910453496627210289749000 \\
&\quad + 3510586678260932028965606400\sqrt{5} \\
&\quad + 2515968\sqrt{3110}(6260208323789001636993322654444020882161 \\
&\qquad + 2799650273060444296577206890718825190235\sqrt{5})^{1/2} \\
C &= -214772995063512240 \\
&\quad - 96049403338648032\sqrt{5} \\
&\quad - 1296\sqrt{5}(10985234579463550323713318473 \\
&\qquad + 4912746253692362754607395912\sqrt{5})^{1/2}
\end{aligned}$$

Then

$$\frac{\sqrt{-C^3}}{\pi} = \sum_{n=0}^{\infty} \frac{(6n)!}{(3n)!(n!)^3} \frac{A+nB}{C^{3n}}$$

Each additional term of the series yields approximately 50 digits. This is an example of a Ramanujan–Sato series.

Cubic convergence (1991) [edit]

Start out by setting

$$a_0 = \frac{1}{3}$$
$$s_0 = \frac{\sqrt{3}-1}{2}$$

Then iterate

$$r_{k+1} = \frac{3}{1+2(1-s_k^3)^{1/3}}$$
$$s_{k+1} = \frac{r_{k+1}-1}{2}$$
$$a_{k+1} = r_{k+1}^2 a_k - 3^k(r_{k+1}^2 - 1)$$

Then a_k converges cubically to 1/π; that is, each iteration approximately triples the number of correct digits.

Another formula for π (1989) [edit]

Start out by setting[citation needed]

$$A = 212175710912\sqrt{61} + 1657145277365$$
$$B = 13773980892672\sqrt{61} + 107578229802750$$
$$C = (5280(236674 + 30303\sqrt{61}))^3$$

Then

$$1/\pi = 12\sum_{n=0}^{\infty} \frac{(-1)^n (6n)!\,(A+nB)}{(n!)^3(3n)!\,C^{n+1/2}}$$

Each additional term of the partial sum yields approximately 31 digits.

Quartic algorithm (1985) [edit]

Start out by setting[1]

$$a_0 = 6 - 4\sqrt{2}$$
$$y_0 = \sqrt{2} - 1$$

Then iterate

$$y_{k+1} = \frac{1-(1-y_k^4)^{1/4}}{1+(1-y_k^4)^{1/4}}$$
$$a_{k+1} = a_k(1+y_{k+1})^4 - 2^{2k+3} y_{k+1}(1 + y_{k+1} + y_{k+1}^2)$$

Then a_k converges quartically against 1/π; that is, each iteration approximately quadruples the number of correct digits.

Quadratic convergence (1984) [edit]

Start out by setting[2] [3]

$$a_0 = \sqrt{2}$$
$$b_0 = 0$$
$$p_0 = 2 + \sqrt{2}$$

Then iterate

$$a_{n+1} = \frac{\sqrt{a_n} + 1/\sqrt{a_n}}{2}$$
$$b_{n+1} = \frac{(1+b_n)\sqrt{a_n}}{a_n + b_n}$$
$$p_{n+1} = \frac{(1+a_{n+1})\,p_n b_{n+1}}{1+b_{n+1}}$$

Then p_k converges quadratically to π; that is, each iteration approximately doubles the number of correct digits. The algorithm is *not* self-correcting; each iteration must be performed with the desired number of correct digits of π.

Quintic convergence [edit]

Start out by setting

$$a_0 = \frac{1}{2}$$
$$s_0 = 5(\sqrt{5} - 2)$$

Then iterate

$$x_{n+1} = \frac{5}{s_n} - 1$$
$$y_{n+1} = (x_{n+1} - 1)^2 + 7$$
$$z_{n+1} = \left(\frac{1}{2}x_{n+1}\left(y_{n+1} + \sqrt{y_{n+1}^2 - 4x_{n+1}^3}\right)\right)^{1/5}$$
$$a_{n+1} = s_n^2 a_n - 5^n\left(\frac{s_n^2 - 5}{2} + \sqrt{s_n(s_n^2 - 2s_n + 5)}\right)$$
$$s_{n+1} = \frac{25}{(z_{n+1} + x_{n+1}/z_{n+1} + 1)^2 s_n}$$

Then a_k converges quintically to 1/π (that is, each iteration approximately quintuples the number of correct digits), and the following condition holds:

$$0 < a_n - \frac{1}{\pi} < 16 \cdot 5^n \cdot e^{-5^n}\pi$$

[4]

Nonic convergence [edit]

Start out by setting

$$a_0 = \frac{1}{3}$$
$$r_0 = \frac{\sqrt{3} - 1}{2}$$
$$s_0 = (1 - r_0^3)^{1/3}$$

Then iterate

$$t_{n+1} = 1 + 2r_n$$
$$u_{n+1} = (9r_n(1 + r_n + r_n^2))^{1/3}$$
$$v_{n+1} = t_{n+1}^2 + t_{n+1}u_{n+1} + u_{n+1}^2$$
$$w_{n+1} = \frac{27(1 + s_n + s_n^2)}{v_{n+1}}$$
$$a_{n+1} = w_{n+1}a_n + 3^{2n-1}(1 - w_{n+1})$$
$$s_{n+1} = \frac{(1 - r_n)^3}{(t_{n+1} + 2u_{n+1})v_{n+1}}$$
$$r_{n+1} = (1 - s_{n+1}^3)^{1/3}$$

Then a_k converges nonically to $1/\pi$; that is, each iteration approximately multiplies the number of correct digits by nine.

See also [edit]

- Gauss–Legendre algorithm – another algorithm to calculate π
- Bailey–Borwein–Plouffe formula

References [edit]

1. ^ Mak, Ronald (2003). *The Java Programmers Guide to Numerical Computation*. Pearson Educational. p. 353. ISBN 0-13-046041-9.
2. ^ Arndt, Jörg; Haenel, Christoph (1998). *π Unleashed*. Springer-Verlag. p. 236. ISBN 3-540-66572-2.
3. ^ Template:Pi Unleashed
4. ^ http://www.cecm.sfu.ca/organics/papers/garvan/paper/html/node12.html

- Pi Formulas from Wolfram MathWorld

Categories: Pi algorithms