The Miraculous Bailey-Borwein-Plouffe Pi Algorithm

Steven Finch, Research and Development Team, MathSoft, Inc.

Overview: ***10/1/95***

David Bailey, Peter Borwein and Simon Plouffe have recently computed the ten billionth digit in the hexadecimal expansion of pi. They utilized an astonishing formula:

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

which enables one to calculate the dth digit of pi without being forced to calculate all the preceding d-1 digits. No one had previously even conjectured that such a *digit-extraction* algorithm for pi was possible. Click here to see the original CECM announcement and here to see a description for a non-technical audience. Bailey, Borwein and Plouffe discovered their formula using the PSLQ lattice reduction algorithm.

We present here a very small verification of the digit-extraction result. Mathcad PLUS 6.0 was useful to us for the purpose of a hasty check up to 1000 digits. A check of more digits is ongoing. We wished to post our work as rapidly as possible, believing that this incomplete work would be of interest to others.

The Mathcad PLUS 6.0 file plffe1.mcd contains a dynamic, working version of our verification. If you have 6.0 and don't know how to view web-based Mathcad files, then you should read these instructions.

The following screenshot exhibits our verification algorithm (which, incidentally, we recognize is *far* less efficient than Bailey, Borwein and Plouffe's original algorithm).

Mathcad PLUS - [PLFFE1.MCD]

File Edit Text Math Graphics Symbolic Window Books Help

$$\mathrm{hexdcm}(a,b,d,e) := \left| \begin{array}{l} x \leftarrow \mathrm{mod}(a,b) \\ y \leftarrow a \ \text{ if } \ d=0 \\ \text{for } k \in 1,2..d \quad \text{otherwise} \\ \quad \left| \begin{array}{l} y \leftarrow 16 \cdot x \\ x \leftarrow \mathrm{mod}(y,b) \end{array} \right. \\ c_0 \leftarrow y - x \\ \text{for } j \in 1,2..e \\ \quad c_j \leftarrow (y \leftarrow 16 \cdot x) - (x \leftarrow \mathrm{mod}(y,b)) \\ \displaystyle\sum_{j=0}^{e} 16^{e-j} \cdot \frac{c_j}{b} \end{array} \right.$$

Verifying the Bailey-Bo
Plouffe digit-extraction

S. Finch, 10/1/95

Hexdcm(a,b,d,e) comput
digits d, d+1, ... , d+e to
right of the decimal point
hexadecimal expansion c
fraction a/b.

Note: e=1 suffices for dig
in the expansion of π wh
$0 \le d \le 10$.
e=2 suffices when $11 \le d$
e=3 suffices when $24 \le d$
e=4 suffices when $47 \le d$

$$\Sigma(d,e) := \sum_{n=0}^{d} \mathrm{hexdcm}\left[120 \cdot n^2 + 151 \cdot n + 47, (8 \cdot n + 1) \cdot (2 \cdot n + 1) \cdot (8 \cdot n + 5) \cdot (4 \cdot n + 3), d - n, e\right]$$

$$\Pi(d,e) := \left| \begin{array}{l} s \leftarrow \mathrm{mod}\left(\Sigma(d,e), 16^{e+1}\right) \\ \dfrac{s - \mathrm{mod}\left(s, 16^{e}\right)}{16^{e}} \end{array} \right.$$

Σ(d,e) sums the (e+1)-di
blocks obtained via hexc
scaled by the appropriat
multiples of 16. Π(d,e) s
away the extraneous dig

auto

$d := 50, 100 .. 1000$

$\Pi(d, 4)$

8h
ch
bh
5h
1h
5h
5h
eh
7h
4h
eh
8h
dh
6h
bh
8h
1h
4h
ah
3h

1000 digits of π, obtained from an independent source:

3.243f6a8885a308d313198a2e03707344a4093822299f31d008
2efa98ec4e6c89452821e638d01377be5466cf34e90c6cc0ac
29b7c97c50dd3f84d5b5b54709179216d5d98979fb1bd1310b
a698dfb5ac2ffd72dbd01adfb7b8e1afed6a267e96ba7c9045
f12c7f9924a19947b3916cf70801f2e2858efc16636920d871
574e69a458fea3f4933d7e0d95748f728eb658718bcd588215
4aee7b54a41dc25a59b59c30d5392af26013c5d1b023286085
f0ca417918b8db38ef8e79dcb0603a180e6c9e0e8bb01e8a3e
d71577c1bd314b2778af2fda55605c60e65525f3aa55ab9457
48986263e8144055ca396a2aab10b6b4cc5c341141e8cea154
86af7c72e993b3ee1411636fbc2a2ba9c55d741831f6ce5c3e
169b87931eafd6ba336c24cf5c7a325381289586773b8f4898
6b4bb9afc4bfe81b6628219361d809ccfb21a991487cac605d
ec8032ef845d5de98575b1dc262302eb651b8823893e81d396
acc50f6d6ff383f442392e0b4482a484200469c8f04a9e1f9b
5e21c66842f6e96c9a670c9c61abd388f06a51a0d2d8542f68
960fa728ab5133a36eef0b6c137a3be4ba3bf0507efb2a98a1
f1651d39af017666ca593e82430e888cee8619456f9fb47d84
a5c33b8b5ebee06f75d885c12073401a449f56c16aa64ed3aa
62363f77061bfedf72429b023d37d0d724d00a1248db0fead3

It is trivial to convert a hexadecimal expansion to a binary expansion. On the other hand, the following question evidently remains unanswered. Do there exist polynomials a(n) and b(n) possessing integer coefficients such that

$$\pi = \sum_{n=0}^{\infty} \frac{a(n)}{b(n)} \cdot \left(\frac{1}{10}\right)^n \qquad ?$$

The just-released, historic Bailey-Borwein-Plouffe paper, CECM preprint P123: On the rapid computation of various polylogarithmic constants starts where we have left off. Reading it, one gains the impression of an emerging and deep theory of transcendental number computation. The untapped consequences of the digit-extraction formula and others like it would appear to be rich and profound.

Postscript: ***1/15/96***

Victor Adamchik and Stan Wagon have recently published a fascinating HTML paper Pi: A 2000-Year Search Changes Direction. This paper presents many new formulas, including the following:

$$\pi = \sum_{k=0}^{\infty} \left(\frac{4+8\cdot r}{8\cdot k+1} - \frac{8\cdot r}{8\cdot k+2} - \frac{4\cdot r}{8\cdot k+3} - \frac{2+8\cdot r}{8\cdot k+4} - \frac{1+2\cdot r}{8\cdot k+5} - \frac{1+2\cdot r}{8\cdot k+6} + \frac{r}{8\cdot k+7} \right) \cdot ($$

for *any* real or complex value r. Observe that this specializes to the original Bailey-Borwein-Plouffe formula when we set r = 0.

We refer the interested reader to the essay Archimedes' constant for a detailed overview of pi facts and formulas and to CECM preprint P130: The Quest for Pi. Plouffe has also compiled a Table of current records for the computation of constants which gives the latest information on calculations such as these.

Postscript: ***10/7/96***

Fabrice Bellard has completed a computation of the 100 billionth hexadecimal digit of pi. Here is his sci.math posting, which interestingly appeared one year, almost to the day, after Bailey, Borwein and Plouffe's announcement. Might the one trillionth hexadecimal digit of pi be known next year?

Postscript: ***1/12/97***

Simon Plouffe has discovered a new algorithm to compute the nth digit of pi and certain other mathematical constants in any base with very little memory. The price of such generality is speed: it is not as fast as the Bailey-Borwein-Plouffe algorithm (but is comparable to other classical methods for computing pi). Here is a description of his work, along with Bellard's improvement.

Postscript: ***2/28/97***

Progress continues! Fabrice Bellard has discovered another miraculous formula which he estimates is 43% faster than the Bailey-Borwein-Plouffe formula. A very simple proof is found here. Another CECM resource is also now available: The Pi Pages.

Postscript: ***9/22/97***

Fabrice Bellard has employed the Bailey-Borwein-Plouffe formula to compute the trillionth ($10^{12\text{th}}$) binary digit of pi. Here is his e-message.

MathSoft | Products | Support | Search
Math in Action | Education | Store | News

This page was modified September 22, 1997