An efficient algorithm for the computation of Bernoulli numbers

Greg Fee
Centre for Constructive and Experimental Mathematics
Simon Fraser University
Vancouver, Canada
gfee@cecm.sfu.ca

Simon Plouffe
Montréal, Canada
mailto:simon.plouffe@gmail.com

Abstract

This article gives a direct formula for the computation of $B(n)$ using the asymptotic formula

$$B(n) \approx 2\,\frac{n!}{\pi^n 2^n}$$

where n is even and $n \gg 1$. This is simply based on the fact that $\zeta(n)$ is very near 1 when n is large and since $B(n) = 2\,\frac{\zeta(n)n!}{\pi^n 2^n}$ exactly. The formula chosen for the Zeta function is the one with prime numbers from the well-known Euler product for $\zeta(n)$. This algorithm is far better than the recurrence formula for the Bernoulli numbers even if each $B(n)$ is computed individually. The author could compute $B(750,000)$ in a few hours. The current record of computation is now (as of Feb. 2007) $B(5,000,000)$ a number of (the numerator) of 27332507 decimal digits is also based on that idea.

1 The need for a single computation

This algorithm came once in 1996 when the authors wanted to compute large Bernoulli numbers using a well-known computer algebra system system like Maple or Mathematica. These programs used Faulhaber's recurrence [2, 5] formula which is nice but unsuitable for large computations. We quickly came to the conclusion that $B(10000)$ was out of reach even with a powerful computer. This is where we realized that for n large the actual formula is simply $B(n) = 2\,\frac{n!}{\pi^n 2^n}$ where n is even and not counting the sign, for n=1000 the approximation is good to more than 300 decimal digits where $B(1000)$ is of the order of 1770 digits. To carry out the exact computation of $B(1000)$ one has only to compute first the principal term in the asymptotic formula and secondly just a few terms in the Euler product (up to $p = 59$). The second idea was that the fractional part of the Bernoulli numbers can also be computed very fast with the help of the Von Staudt-Clausen formula. So finally, the need is only to compute B_n with enough precision so that the remainder is < 1 and apply the Von Staudt-Clausen formula for the fractional part to finally add the 2 results. Note : Mathematica now uses a much more efficient algorithm partly due to these results presented here.

2 The Von Staudt-Clausen formula

The formula is, for $k \geq 1$,

$$(-1)^k B_{2k} \equiv \sum \left(\frac{1}{p}\right) \mod 1.$$

The sum being extended over primes p such that $(p-1)|2k$ [5]. In other words, for $B(10)$ the sum is

$$B(10) = 1 - 1/2 - 1/3 - 1/11 = 5/66.$$

In terms of computation, when n is of the order of 1000000 it goes very fast to compute the fractional part of B_n. The only thing that remains to be done then is the principal part or integer part of B_n.

3 The Euler product

The Euler product of the zeta function is

$$\sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \in \mathbb{P}} \frac{1}{1-p^{-s}}.$$

Where $s > 1$ and p is prime. This is the error term in B_n. For any given n there are $\frac{n}{\ln(n)}$ primes compared to n. Translated into the program it means less operations to carry, the program stops when p^k is of the order of $B\left(n\right)$.

4 The final program

The Maple program uses a high precision value of 2π and a routine for the Von Staudt-Clausen formula. That program held the record of the computation of Bernoulli Numbers from 1996 to 2002, after that others made more efficient programs using C++ and high precision packages like Kellner and Pavlyk (see table 1) and could reach $B(5,000,000)$.
The program was used in 2003 to verify Agoh's conjecture up to n=49999 by the authors. Agoh's conjecture is

$$pB_{p-1} \equiv -1 \mod p$$

is true iff p is prime. The congruence is not obvious since pB_{p-1} is a fraction. The standard method reduces first the numerator mod p, then re-evaluates the fraction, then reduces the numerator mod p. The final fraction is always smaller than 1 and the result of $a/b \mod p$ is solved by finding k such that $a \equiv bk \mod p$. There are 3 parts in the main program which may take time. First the computation of $(2\pi)^n$ and $n!$. Secondly, the evaluation of the Von Staudt-Clausen formula and thirdly the computation of the Euler product. On a medium sized computer (Pentium 2.4 Ghz with Maple 10 and 1 gigabyte of memory). The run time for $B(20000)$ is about 9 seconds and the number is 61382 digits long including 1 second to read the value of π to high-precision from the disk. Here are the timings for that run :

- Product with primes up to 1181 at 61382 digits of precision : 7 seconds.
- Exponentiation of 2π and $n!$: less than 1 second.
- Computation of 20000! : negligible.
- Computation of Von Staudt-Clausen expression : negligible.

When n increases the time taken to evaluate the product with primes is what takes the most. A value of π to several thousands digits is necessary. Maple can supply many thousands but a file containing 1 million is easily found on the internet and is much faster. In this program π is renamed pi with no capitals. The Bernoulli numbers up to $n = 100$ are within the program mainly for speed when n is small.

```
BERN:=proc(n::integer)
local d, z, oz, i, p, pn, pn1, f, s, p1, t1, t2;
global Digits;
    lprint(‘start at time‘ = time());
    if n = 1 then -1/2
    elif n = 0 then 1
    elif n < 0 then ERROR(‘argument must be >= 0‘)
    elif irem(n, 2) = 1 then 0
    elif n <= 100 then op(iquo(n, 2), [1/6, -1/30, 1/42, -1/30,
        5/66, -691/2730, 7/6, -3617/510, 43867/798, -174611/330,
        854513/138, -236364091/2730, 8553103/6, -23749461029/870,
        8615841276005/14322, -7709321041217/510, 2577687858367/6,
        -26315271553053477373/1919190, 2929993913841559/6,
        -261082718496449122051/13530, 1520097643918070802691/1806,
        -27833269579301024235023/690, 596451111593912163277961/282,
        -5609403368997817686249127547/46410,
        495057205241079648212477525/66,
        -801165718135489957347924991853/1590,
```

```
    29149963634884862421418123812691/798,
    -2479392929313226753685415739663229/870,
    84483613348880041862046775994036021/354,
    -1215233140483755572040304994079820246041491/56786730,
    12300585434086858541953039857403386151/6,
    -106783830147866529886385444979142647942017/510,
    1472600022126335654051619428551932342241899101/64722,
    -78773130858718728141909149208474606244347001/30,
    1505381347333367003803076567377857208511438160235/4686,
    -5827954961669944110438277244641067365282488301844260429/
    140100870,
    34152417289221168014330073731472635186688307783087/6,
    -24655088825935372707687196040585199904365267828865801/30,
    414846365575400828295179035549542073492199375372400483487/
    3318, -4603784299479457646935574969019046849794257872751 2\
    88919656867/230010, 1677014149185145836823154509786269900\
    20773602757025341488161 3/498, -2024576195935290360231131 1\
    601117310099899173911980908772810839324 77/3404310, 660714\
    619417678653573847847426261496277830686653388931761996 98\
    3/6, -131142648867401750799551142401931184334575027557202\
    86442969198905740 47/61410, 117905727902108279988412335124\
    921508377525494966964711623154521572792253 5/272118, -1295\
    585948207537527989427828538576749659341483719435143023 31\
    6326829946247/1410, 122081380657974446960730167941320120 3\
    958508415202696621436215105284649447/6, -2116004495972665\
    130975977281098242336730439543890602341506387334200506 68\
    349987259/4501770, 679082606729054956240511175464036056 07\
    342195728504487509073961249992947058239/6, -9459803781912\
    212529522743306949372187270284153306693613338569620431 13\
    95415197247711/33330])
else
    d := 4
         + trunc(evalhf((lnGAMMA(n + 1) - n*ln(2*Pi))/ln(10)))
         + length(n);
    lprint(`using ` . d . ` Digits`);
    s := trunc(evalhf(exp(0.5*d*ln(10)/n))) + 1;
    Digits := d;
    p := 1;
    t1 := 1.;
    t2 := t1;
```

```
lprint(`start small prime loop at time` = time());
while p <= s do
    p := nextprime(p);
    pn := p^n;
    pn1 := pn - 1;
    t1 := pn*t1;
    t2 := pn1*t2
end do;
gc();
lprint(status);
lprint(`used primes up to and including ` . p);
lprint(`finish small prime loop at time` = time());
z := t1/t2;
gc();
lprint(status);
lprint(`finish full prec. division at time` = time());
oz := 0;
while oz <> z do
    oz := z;
    p := nextprime(p);
    Digits := max(d - ilog10(pn), 9);
    pn := Float(p,0);
    pn := p^n;
    pn1 := z/pn;
    Digits := d;
    z := z + pn1
end do;
gc();
lprint(status);
lprint(`used primes up to and including ` . p);
lprint(`finish big prime loop at time` = time());
p := evalf(2*pi);
gc();
lprint(status);
lprint(`finish 2*Pi at time` = time());
f := n!;
gc();
lprint(status);
lprint(`finish factorial at time` = time());
pn := p^n;
```

```
        gc();
        lprint(status);
        lprint(`finish (2*Pi)^n at time` = time());
        z := 2*z*f/pn;
        gc();
        lprint(status);
        lprint(
            `finish 2*z*n!/(2*Pi)^n (multiply and divide) at time`
             = time());
        s := 0;
        for p in numtheory[divisors](n) do
            p1 := p + 1; if isprime(p1) then s := s + 1/p1 end if
        end do;
        gc();
        lprint(status);
        lprint(`finish divisors of n loop at time` = time());
        s := frac(s);
        if irem(n, 4) = 0 then
            if s < 1/2 then z := -round(z) - s
            else z := -trunc(z) - s
            end if
        else
            s := 1 - s;
            if s < 1/2 then z := round(z) + s
            else z := trunc(z) + s
            end if
        end if;
        gc();
        lprint(status);
        lprint(`done at time` = time());
        z
    end if
end:
```

Who	when	highest B_n
Bernoulli	1713	10
Euler	1748	30
J.C. Adams	1878	62
D.E. Knuth and Buckholtz	1967	360
Greg Fee and Simon Plouffe	1996	10000
Greg Fee and Simon Plouffe	1996	20000
Greg Fee and Simon Plouffe	1996	30000
Greg Fee and Simon Plouffe	1996	50000
Greg Fee and Simon Plouffe	1996	100000
Greg Fee and Simon Plouffe	1996	200000
Simon Plouffe	2001	250000
Simon Plouffe	2002	400000
Simon Plouffe	2002	500000
Simon Plouffe	2002	750000
Berndt C. Kellner	2002	1000000
Berndt C. Kellner	2003	2000000
Pavlyk O.	2005	5000000

Table 1: History of the computation of Bernoulli numbers

References

[1] N. J. A. Sloane, "The On-Line Encyclopedia of Integer Sequences," available at http://www.research.att.com/∼njas/sequences/.

[2] M. Abramowitz and I. Stegun, editors. *Handbook of Mathematical Functions*. Dover, New York, 1970.

[3] K. Dilcher, , *A Bibliography of Bernoulli Numbers*, http://www.mscs.dal.ca/ dilcher/bernoulli.html.

[4] Xavier Gourdon et Patrick Sebah, , *Introduction to Bernoulli Numbers*, http://numbers.computation.free.fr/Constants/Miscellaneous/bernoulli.html.

[5] G. H. Hardy and E. L. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, 1979.

[6] Kellner, B. C. *ber irregulre Paare hherer Ordnungen*. Diplomarbeit. Gttingen, Germany: Mathematischen Institut der Georg August Universitt zu Gttingen, 2002. http://www.bernoulli.org/ bk/irrpairord.pdf.

[7] Kellner, B. C. *The Equivalence of Giuga's and Agoh's Conjectures*. 15 Sep 2004. http://arxiv.org/abs/math.NT/0409259/.

[8] D. E. Knuth, *The Art of Computer Programming* vol. 2, Addison-Wesley, Reading, MA, 1981 and 1997.

[9] S. Plouffe, *The computation of big Bernoulli numbers*, from 1996 to 2002, on the author web site at http://www.lacim.uqam.ca/∼plouffe/Bigfiles/.

[10] S. Ramanujan, *Some properties of Bernoulli's numbers*, J. Indian Math. Soc., (1911), vol. 3, pp. 219-234.

[11] P. Ribenboim, *The new Book of Prime Number Records*, Springer, (1996).

[12] Neil J.A. Sloane and Simon Plouffe,, *The Encyclopedia of Integer Sequences*, Academic Press, San Diego 1995, 587 pp.

[13] Neil Sloane and al. *The On-Line Encyclopedia of Integer Sequences*, available at http://www.research.att.com/ njas/sequences/.

[14] Eric Weisstein, *Agoh's conjecture*, http://mathworld.wolfram.com/AgohsConjecture.html.

[15] Eric Weisstein, *Bernoulli Numbers*, http://mathworld.wolfram.com/BernoulliNumber.html.

2000 *Mathematics Subject Classification*: Primary 11B68; Secondary 05A10, 11A07, 11B64.
Keywords: Bernoulli numbers, Euler product, Zeta function.

(Concerned with sequences A000367, A000928, A000928, A002445, A027641, A027642)

Return to Journal of Integer Sequences home page.