Calcbn - A program for calculating Bernoulli numbers

Bernd C. Kellner

16th May 2004

Abstract

This manual gives a short introduction to Bernoulli numbers and a description of the program **Calcbn** V1.2 which calculates Bernoulli numbers in a fast way and permits several options.

Keywords: Bernoulli numbers

Mathematics Subject Classification 2000: 11B68

1 Introduction

The Bernoulli numbers B_n play an important role in several topics of mathematics. These numbers can be defined by the power series

$$\frac{z}{e^z - 1} = \sum_{n=0}^{\infty} B_n \frac{z^n}{n!}\,, \qquad |z| < 2\pi\,,$$

where all numbers B_n are zero with odd index $n > 1$. The even-indexed numbers alternate in sign. First values are given by the following table.

n	0	1	2	4	6	8	10	12	14	16	18	20
B_n	1	$-\frac{1}{2}$	$\frac{1}{6}$	$-\frac{1}{30}$	$\frac{1}{42}$	$-\frac{1}{30}$	$\frac{5}{66}$	$-\frac{691}{2730}$	$\frac{7}{6}$	$-\frac{3617}{510}$	$\frac{43867}{798}$	$-\frac{174611}{330}$

One also has a recursive formula

$$B_n = -\frac{1}{n+1} \sum_{k=0}^{n-1} \binom{n+1}{k} B_k\,, \quad n \geq 1\,. \tag{1.1}$$

The Bernoulli numbers are intimately connected with the Riemann zeta function

$$\zeta(s) = \sum_{\nu=1}^{\infty} \nu^{-s} = \prod_p (1 - p^{-s})^{-1}\,, \quad s \in \mathbb{C}\,,\ \mathrm{Re}\, s > 1\,, \tag{1.2}$$

by Euler's formula for even positive integers n

$$\zeta(n) = -\frac{1}{2} \frac{(2\pi i)^n}{n!} B_n\,, \quad n \in \mathbb{N}\,,\ 2 \mid n\,. \tag{1.3}$$

Since $\zeta(n) \to 1$ for $n \to \infty$, (1.3) shows the rapidly growing of Bernoulli numbers. The calculation of these numbers can be done recursively for small indices by (1.1). But for higher indices say $n > 1\,000$ or even $n = 1\,000\,000$, the classical recursive formula (1.1) breaks down, because we then need for each step all calculated numbers before. Now, equation (1.3) allows the direct calculation of Bernoulli numbers, since the calculation of π to arbitrary precision is no problem nowadays, see [BBBP97] for new algorithms.

The algorithm of **Calcbn** initially calculates an approximation of the numerator of B_n, then the main part of calculation is done using integers only via Euler factors, see [Kel02, Section 2.7] for details. Also the C++ source code of **Calcbn** is available, see [Kel02, Appendix B, pp. 134]. Note that the current version 1.2 of **Calcbn** has been improved by faster calculation of factorials and powers and further optimizations. In another paper the new methods will be described soon.

The program **Calcbn** [Kel03] uses the very usable C++ High Performance Arbitrary Precision Arithmetic Package **apfloat** of Tommila [Tom01] which allows an easy implementation of the algorithm. The current version of **Calcbn** can calculate Bernoulli numbers up to index $n = 1\,000\,000$. For example, the calculation of $B_{20\,000}$ takes 17 seconds on a 1.5 GHz Pentium M with 512 MB RAM.

To understand the structure of Bernoulli numbers better, we present some well-known properties, see [IR90, Chapter 15]; some options of **Calcbn** are dealing with.

Theorem 1.1 (Clausen-von Staudt) *Let n be an even positive integer. Then*

$$B_n + \sum_{p-1|n} \frac{1}{p} \in \mathbb{Z} \qquad \textit{and} \qquad \mathrm{denom}(B_n) = \prod_{p-1|n} p\,.$$

Thus, the denominator of B_n is given easily. The continuation of the Riemann zeta function to the whole complex plane yields

$$\zeta(1-n) = -\frac{B_n}{n}\,, \quad n \in \mathbb{N}\,, \quad n \geq 2\,.$$

The divided Bernoulli numbers B_n/n have very interesting properties.

Theorem 1.2 (Kummer) *Let φ be the Euler φ-function, then the Kummer congruences state for $n, m, p, r \in \mathbb{N}$, n, m even, p prime and $p - 1 \nmid n$*

$$(1 - p^{n-1})\frac{B_n}{n} \equiv (1 - p^{m-1})\frac{B_m}{m} \pmod{p^r} \tag{1.4}$$

with $n \equiv m \pmod{\varphi(p^r)}$.

Now, these important congruences especially provide the property

$$p^r \mid \frac{B_n}{n} \quad \Longleftrightarrow \quad p^r \mid \frac{B_{n+k\varphi(p^r)}}{n+k\varphi(p^r)} \quad \text{for all } k \geq 0 \tag{1.5}$$

with $2 \leq n < \varphi(p^r) = p^{r-1}(p-1)$. The primes which occur in the numerator of B_n/n are the so-called irregular primes. The occurrence of irregular primes p is described by the definition of irregular pairs. Now, the behavior of irregular prime powers is principally clarified giving the whole structure of the Bernoulli number B_n as well of B_n/n, see [Kel04] for details.

2 Program Calcbn

The program **Calcbn** runs under Windows (32Bit) and Linux (32Bit). **Calcbn** can be freely used without warranty of any kind, all rights reserved. All input options are given via command line and outputs are printed into the console.

2.1 Command line options

`>calcbn` ↩ prints the command line options.

```
Usage: calcbn [-opt] start [end] [step]
  -c: copyright and version

  default: print factors, '.' for incomplete factorization
  -p num: factorization up to prime <= num (default = 500000)
  -t: report time of calculation
  -z: calculate Bn/n
  -Z: splitted numerator of n and Bn/n

  -s: signed number (default)
  -u: unsigned number
  -n: print numerator
  -N: print numerator/denominator

  -d num:     divide by num
  -D num pow: divide by num^pow
  -m num:     mod num
  -M num pow: mod num^pow
  -i calculate inverse modulo
  -r prime pow: step = phi(prime^pow)
  -R prime pow: -r prime pow  and  -D prime pow

  --m0: method with summation
  --m1: method with Euler factors (default)
  --check num: check result with num different prime moduli

  start: even start index (max index = 1000000)
  end:   even end index (default = start)
  step:  even step (default = 2)
```

2.2 Factorization of the numerator

The default options provide the factorization of numerators of B_n without sign. The factors are given up to primes less 500 000 except the remaining factor is recognized as a prime. The Option `-p 1000000` raises the factorization to the limit. In case of an incomplete factorization a dot is printed at the end.

`>calcbn 10 30 4` ↩ Numerators of B_n

```
10: 5
14: 7
18: 43867
22: 11 131 593
26: 13 657931
30: 5 1721 1001259881
```

`>calcbn -z 10 30 4` ↩ Numerators of B_n/n

```
10: 1
14: 1
18: 43867
22: 131 593
26: 657931
30: 1721 1001259881
```

`>calcbn -Z 10 30 4` ↩ Numerators of B_n and B_n/n, factors are partitioned

```
10: 5 * 1
14: 7 * 1
18: 1 * 43867
22: 11 * 131 593
26: 13 * 657931
30: 5 * 1721 1001259881
```

Note that the left side gives a trivial divisor of the numerator of B_n which can be determined easily. The right side after $*$ consists only of irregular primes.

2.3 Calculation of numerators and denominators

`>calcbn -n 10 20` ↩ Numerators of B_n

```
10: 5
12: -691
14: 7
16: -3617
18: 43867
20: -174611
```

`>calcbn -N 10 20` ↩ B_n as a rational number

```
10: 5/66
12: -691/2730
14: 7/6
16: -3617/510
18: 43867/798
20: -174611/330
```

The additional option `-z` yields the corresponding calculation of B_n/n. The option `-s` which is default resp. option `-u` supplies signed resp. unsigned numbers.

2.4 Modulo arithmetics

Several options allow the calculation of Bernoulli numbers B_n resp. B_n/n in modulo arithmetics. These options are required by calculations related to Kummer congruences (1.4) and property (1.5). Let B_n^* denote B_n (default) resp. B_n/n (option `-z`). Let $B_n^* = U/V$ with $(U,V)=1$, $V>0$.

Option	Result
`-d` m	divides B_n^* by m. If not possible '?d' is printed.
`-D` m r	divides B_n^* by m^r. If not possible '?d' is printed.
`-m` m	$B_n^* \pmod{m}$
`-M` m r	$B_n^* \pmod{m^r}$
`-i`	enables also `-N` and needs `-m` / `-M`.
	Then UV^{-1} (mod) is calculated otherwise the fraction U/V (mod).
`-r` p r	sets step to the value $\varphi(p^r) = p^{r-1}(p-1)$
`-R` p r	enables also `-r` and divides B_n^* by p^r

Digits of numerator

The last decimal digits of the unsigned numerator are given as follows. For example, the numerator of $B_{25\,000}$ ends with the following 6 decimal digits.

```
>calcbn -n -u -M 10 6 25000 ↩
```

```
25000: 178901 (mod 10^6)
```

Kummer congruences

Let $p = 13$, $r = 3$, $n = 20$, $m = 2048$, $\varphi(p^r) = 2028$. Then

$$(1 - p^{n-1}) \frac{B_n}{n} \equiv (1 - p^{m-1}) \frac{B_m}{m} \pmod{p^r}$$

is easily checked since the Euler factors here vanish.

```
>calcbn -z -N -M 13 3 -i -r 13 3 20 2048 ↩
```

```
20: 860 (mod 13^3)
2048: 860 (mod 13^3)
```

Sequences related to irregular prime powers

Here we will calculate the following sequence

$$\frac{B_{n_\nu}}{p^2\, n_\nu} \pmod{p^5}$$

with $n_\nu = n + \nu\varphi(p^2)$ and requiring $p^2 \mid B_n/n$ by (1.5). We can take $p = 37$, $n = 284$.

`>calcbn -z 284 ↩` Checking factorization of numerator of B_n/n

```
284: 37^2 .
```

`>calcbn -z -N -M 37 5 -i -D 37 2 284 ↩` One value

```
284: 5099233 (mod 37^5)
```

`>calcbn -z -N -M 37 5 -i -R 37 2 284 6944 ↩` Sequence

```
284: 5099233 (mod 37^5)
1616: 51897779 (mod 37^5)
2948: 69317585 (mod 37^5)
4280: 25497914 (mod 37^5)
5612: 27265943 (mod 37^5)
6944: 42760935 (mod 37^5)
```

2.5 General options

Option	Result
`-c`	prints copyright and version.
`-t`	reports the time of each calculation in brackets [ ] and the total time in sec.
`--m0`	calculates B_n via a partial sum of $\zeta(n)$ (1.2) which is slower.
`--m1`	calculates B_n via Euler factors (1.2) using integers which is default.
`--check` m	verifies B_n via Kummer congruences (1.4) with m different primes p.

References

[BBBP97] D. H. Bailey, J. M. Borwein, P. B. Borwein, and S. Plouffe. The Quest for Pi. *Mathematical Intelligencer*, 19:50–57, 1997.

[IR90] K. Ireland and M. Rosen. *A Classical Introduction to Modern Number Theory*, volume 84 of *Graduate Texts in Mathematics*. Springer-Verlag, 2nd edition, 1990.

[Kel02] B. C. Kellner. Über irreguläre Paare höherer Ordnungen. *Diplomarbeit. Mathematisches Institut der Georg August Universität zu Göttingen, Germany. Also available at http://www.bernoulli.org/∼bk/irrpairord.pdf*, 2002.

[Kel03] B. C. Kellner. Calcbn. A program for calculating Bernoulli numbers. *Available at http://www.bernoulli.org/*, 2003.

[Kel04] B. C. Kellner. On irregular prime powers of Bernoulli numbers. *Preprint, to appear. Also available at http://www.bernoulli.org/∼bk/irrpowbn.pdf*, pages 1–29, 2004.

[Tom01] M. Tommila. A C++ High Performance Arbitrary Precision Arithmetic Package, apfloat 2.33. *http://www.apfloat.org*, September 2001.

Bernd C. Kellner
address: Reitstallstr. 7, 37073 Göttingen, Germany
email: bk@bernoulli.org