ベルヌーイ数 B_n を求めるプログラム

金沢大学４年　数物化学類

荒井　雄太

・プログラム作成の経緯と結果

４年次の数学講究で扱っていた「ベルヌーイ数とゼータ関数」(発行：牧野書店) の P.35 で、ベルヌーイ数を求めるためのアルゴリズムが掲載されており、実際にプログラムとして作成してみようということに至った。プログラムを作成し実際に動かしてみた結果、n=16 までの数であれば正確に分母、分子に分けて B_n を表示できるプログラムを作成できた。n=17 以降では誤差が発生しており、その理由は後述する。

・元にしたアルゴリズム

$a_{0,m}(m=0,1,2,\cdot\cdot\cdot)$ が与えられていて、$a_{n,m}(n\geq 1)$ が

$a_{n,m}=(m+1)(a_{n-1,m}-a_{n-1,m+1})\quad(n\geq 1, m\geq 0)$ で定められているとき、

$$a_{n,0}=\sum_{m=0}^{n}(-1)^m m!\left\{ {n+1 \atop m+1} \right\}a_{0,m}\quad(n\geq 0)$$

({　}部は第２種ベルヌーイ数)

また、

$$B_n=\sum_{m=0}^{n}\frac{(-1)^m m!\left\{ {n+1 \atop m+1} \right\}}{m+1}\quad(n\geq 0)$$

より、

第 0 行目に $1, \frac{1}{2}, \frac{1}{3}, \cdot\cdot\cdot, \frac{1}{m+1}, \cdot\cdot\cdot$ と初期値を置き、以下第 n 行目$(n\geq 1)$の左から m 番目の数を $a_{n,m}$ とするとき、第 n+1 行目の左から m 番目の数 $a_{n+1,m+1}$ を

$a_{n+1,m}=(m+1)(a_{n,m}-a_{n,m+1})\quad(n\geq 1, m\geq 0)$

で決めていったとき、各行の先頭に現れる数 $a_{n,0}$ はベルヌーイ数 B_n と等しいという性質を利用してプログラムを作成した。

・作成したプログラム

```
program bernoulli
integer n
parameter(n=20)                          B0～Bn までのベルヌーイ数
                                          を求めていく
integer m,s,i
integer a1(0:n),a2(0:n),b1(0:n),b2(0:n)  a1、a2 を初期値が入る箱、
                                          b1、b2 を求めた Bn を入れる
do m=0,n                                 箱として使う
   a1(m)=1                               (a1、b1 には分子の数が、
```

```
   a2(m)=m+1
   b1(m)=0
   b2(m)=0
enddo
```

a2、b2 には分母の数が入る)

最初、$a_{0,m}$ には初期値として
$\frac{1}{m+1}$ が入るので、a1(m)には 1、
a2(m)には m+1 が入る($0 \leq m \leq n$)

```
b1(0)=a1(0)
b2(0)=a2(0)
```

$B_0=a_{0,0}$ なので、b1(0)=a1(0)、
b2(0)=a2(0)となる

以下、n=1 以降の B_n を求めていく

```
do s=1,n
   do m=0,n-s
```

s 行目を求める計算を以下で行う
s 行目の m 番目の値の計算を行う

$a_{n+1,m}=(m+1)(a_{n,m}-a_{n,m+1})$ より、
$a_{s,m}$ を求めるには $a_{s-1,m}$、$a_{s-1,m+1}$ の
2 つの値が分かっていればよい。

$$(m+1)(a_{s-1,m}-a_{s-1,m+1})=(m+1)\left(\frac{a1(m)}{a2(m)}-\frac{a1(m+1)}{a2(m+1)}\right)$$

$$=\frac{(m+1)(a1(m)*a2(m+1)-a2(m)*a1(m+1))}{a2(m)*a2(m+1)}$$

となるので、a1(m),a2(m)の値を
入れ直していく。

```
      a1(m)=(m+1)*(a1(m)*a2(m+1)-a2(m)*a1(m+1))
      a2(m)=a2(m)*a2(m+1)
```

```
   if((a1(m).lt.0).and.(a2(m).lt.0)) then
      a1(m)=abs(a1(m))
      a2(m)=abs(a2(m))
   endif
```

計算の結果、分子、分母共に
負の符号がついたときは
両方とも正の符号に通分する

```
   if((a1(m).ge.0).and.(a2(m).lt.0)) then
      a1(m)=-a1(m)
      a2(m)=abs(a2(m))
   endif
```

分母のみに負の符号がついた
ときは、分子に負の符号をつけ、
分母は正の符号の数に直す
このとき、a1 と a2 は通分されて

```
                                                  いないので、以下で通分していく
          call euclid(abs(a1(m)), abs(a2(m)), i)
                                                  ユークリッドの互除法を用いて
                                                  |a1|と|a2|の最大公約数 i を
                                                   求める
          a1(m)=a1(m)/i                           分子と分母から i を割って
          a2(m)=a2(m)/i                           通分完了
       enddo

       b1(s)=a1(0)                                Bs,0=as,0 なので、a1(0)と a2(0)
       b2(s)=a2(0)                                を b1(0)と b2(0)に入力する

       if (b1(s) .eq. 0) then
          b2(s)=0                                 結果表示のとき、分子が 0 ならば
       endif                                      分母も 0 と表示されるようにする

    enddo                                         上に戻って s+1 行目を求める計算
                                                  を行う

    do s=0,n
       write(*,*) 'n=' ,s,b1(s),b2(s)             s=0 から n まで、s,Bs の分子,分母
    enddo                                         の順番で計算結果をリスト表示

    end                                          プログラム本体はここまでで終了

c------------------------------          以下は、gcd を求めるサブルーチンプログラム

    subroutine euclid(a, b, s)                   |a|と|b|の最大公約数 s を求める
    integer n
    parameter(n=500)                             十分大きな数(ここでは 500)を
    integer a,b,s                                 n として置く
    integer i,k
    integer r(0:n)

    s=0
    if (a .lt. 0) then
       a=abs(a)                                  a,b が負の数のときは正の数に直す
```

```
      endif

      if (b .lt. 0) then
         b=abs(b)
      endif

      do i=0,n
         r(i)=0
      enddo

      r(0)=a
      r(1)=b                                   r(0)=a,r(1)=b とし、以下、
                                                r(k)=mod(r(k－2),r(k－1))
      k=0                                      を、r(k)=0 が出るまで求めていく
 10   if (r(k+1).ne.0)then                     r(k)=0 が出たら、r(k-1)が求めたい
         k=k+1                                  最大公約数なので、その値を s として
         r(k+1)=mod(r(k-1),r(k))               計算終了
         go to 10
      endif

      s=r(k)
      end                                      ここまででサブルーチン部分は終了
```

実行結果

```
n=            0           1           1
 n=           1           1           2
 n=           2           1           6
 n=           3           0           0
 n=           4          -1          30
 n=           5           0           0
 n=           6           1          42
 n=           7           0           0
 n=           8          -1          30
 n=           9           0           0
```

n=	10	5	66
n=	11	0	0
n=	12	-691	2730
n=	13	0	0
n=	14	7	6
n=	15	0	0
n=	16	-3617	510
n=	17	268435456	299505991
n=	18	380323201	353564858
n=	19	-268435456	166710743
n=	20	-240748051	597494090

・問題点

n=17 以降から誤差が出始めている

・n=17 以降誤差が出始めている理由

以下は途中計算で a1(m)、a2(m)の値がどう変わっていったかを表示したグラフの一部(左から n、m、a1(m) ($a_{n,m}$ の分子)、a2(m) ($a_{n,m}$ の分母)の順)

10	0	5	66
10	1	5	66
10	2	-1017	20020
10	3	-2663	15015
10	4	-35	143
10	5	-1013	4004
10	6	-38759	175032
10	7	-18536	109395
11	0	0	1
11	1	691	2730
11	2	691	1820
11	3	368	1365
11	4	15	364
11	5	-3515	18564
11	6	1620401021	1967756456
12	0	-691	2730

ここで、$a_{10,6}$ と $a_{10,7}$ から $a_{11,6}$ を求めるときに、$a_{10,6}$ と $a_{10,7}$ の分母の積は
175032×109395=19147625640 なのに対し、$a_{11,6}$ の分母は 1967756456 となっていて、
19147625640/1967756456=9.730688765683308 から、分母が上手く通分されていないことが分かる。$a_{11,6}$ が $a_{n,0}$ を求める際に使われる数となるのが n=17 からなので、
n=17 以降から誤差が出始めている。

・整数同士の計算に誤差が出ている理由
もともとプログラムで計算を行うとき、CPU が直接扱える数の範囲には限度があり、
今回の場合だと $2^{32}=4294967296$ 種類までの数しか扱えないという制限がある。
$a_{11,6}$ の途中計算でこの限度を上回る数が出てきたので、限度に納まるような数に途中で変更されていたので、計算に誤差が出ていました。

・参考文献
ベルヌーイ数とゼータ関数(発行：牧野書店　著：荒川恒男、伊吹山知義、金子昌信)