数値計算の誤差

０．目次

１．情報落ち

計算のルールを「10進4桁、切り捨て」と仮定する。
２つの数の加算では、まず小数点が合わされ、大きい数が優先される。
したがって、12.34 + 0.005678 は 12.34 と計算される。このように、
絶対値の小さい数を絶対値の大きい数に加えてもほとんど影響を
与えない現象を情報落ちという。

２．オーバーフロー・アンダーフロー

計算結果の絶対値がコンピュータの処理できる最大の数を越えてしまう
現象をオーバーフローという。

計算結果の絶対値がコンピュータの処理できる最小数以下になる現象を
いう。アンダーフローが生じると計算結果を０にすることが多い。

３．桁落ち

12.34 - 12.33 = 0.01 のように、同符号で絶対値のほぼ等しい２個の
数に対して、減算を行うと有効数字が著しく減ってしまう現象をいう。

４．丸め誤差の累積

コンピュータの内部で処理できる数値の桁数が決まっているため、
丸め（４捨５入、切り捨て、切り上げ）が起こる。このために生じる
誤差を丸め誤差という。

１．情報落ち

計算のルールを「10進4桁、切り捨て」と仮定する。２つの数の加算では、まず小数点が合わされ、大きい数が優先される。したがって、12.34 + 0.005678 は 12.34 と計算される。

	1	2	.	3	4				
＋		0	.	0	0	5	6	7	8
	1	2	.	3	4				

このように、絶対値の小さい数を絶対値の大きい数に加えてもほとんど影響を与えない現象を**情報落ち**という。

10**nに1を加え情報落ちが発生したら表示する。

●プログラム(MA111.bas)

```
1  ' << MA111.bas >>
2  ' 情報落ち
3  '
4  ' 数値変数の場合。*/
5  A=10^10
6  For N=11 To 20
7    A=A*10
8    B=A+1
9    Print"10^";N;"+1=";B;
10   If A = B Then Print"  情報落ち発生";
11   Print
12 Next N
13 End
```

実行結果

```
1  10^ 11+1=  100000000001
2  10^ 12+1=  1000000000001
3  10^ 13+1=  10000000000001
4  10^ 14+1=  100000000000001
5  10^ 15+1=  1000000000000001
6  10^ 16+1=  10000000000000001
7  10^ 17+1=  100000000000000001
8  10^ 18+1=  1E18
9  10^ 19+1=  1E19
10 10^ 20+1=  1E20 情報落ち発生
11 OK
```

●プログラム(MA112.bas)

```
1  ' << MA112.bas >>
2  ' 情報落ち
3  '
4  ' 数値変数の場合。*/
5  A=2^50
6  For N=51 To 65
7    A=A*2
8    B=A+1
9    Print"2^";N;"+1=";B;
10   If A = B Then Print"  情報落ち発生";
11   Print
12 Next N
13 End
```

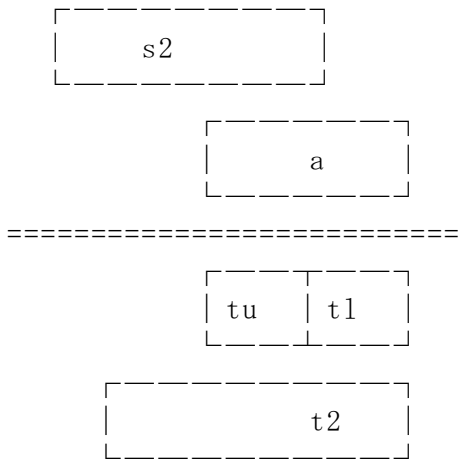
実行結果

```
1  2^ 51+1= 2251799813685249
2  2^ 52+1= 4503599627370497
3  2^ 53+1= 9007199254740993
4  2^ 54+1= 18014398509481985
5  2^ 55+1= 36028797018963969
6  2^ 56+1= 72057594037927937
7  2^ 57+1= 144115188075855873
8  2^ 58+1= 288230376151711745
9  2^ 59+1= 576460752303423489
10 2^ 60+1= 1.15292150460684698E18
11 2^ 61+1= 2.30584300921369395E18
12 2^ 62+1= 4.6116860184273879E18
13 2^ 63+1= 9.22337203685477581E18
14 2^ 64+1= 1.84467440737095516E19  情報落ち発生
15 2^ 65+1= 3.68934881474191032E19  情報落ち発生
16 OK
```

「定数aをn回加える」操作を３つの方法で比較する。

方法１：順にaをn回加える。情報落ちが発生する。

方法２：順にaをn回加えるが、情報落ちをできるだけ防ぐ工夫をする。



tlは情報落ちのため加算に反映されない部分

すなわち、aをtuとtlに分割し、tuはs2に加え、tlはt2に加える。
このようにすると、情報落ちで捨てられるtlがt2に保存され、
情報落ちをある程度防ぐことができる。

●プログラム(MA121.bas)

```
1  ' << MA121.bas >>
2  ' 情報落ちとその改善
3  '
4  Do
5    ' AとNの読み込み。
6    Read A,N
7    If (A = 0) or (N = 0) Then Exit Do
8    Print"A=";A;" N=";N;" N*A=";N*A
9    '
10   ' 方法１：順に加えていく方法。情報落ちが生じる。
11   S1=0.0
12   For I=1 To N
13     S1=S1+A
14   Next I
15   Print"方法１：";S1
16   '
17   ' 方法２
18   S2=0.0: T2=0.0
19   For I=1 To N
20     W=S2+A
21     TU=W-S2
22     TL=A-TU
23     T2=T2+TL
24     S2=S2+TU
25   Next I
```

```
26    Print"方法２：";S2+T2
27  Loop
28  End
29  '
30  ' データ。
31  Data 1.0000000000001, 1000
32  Data 1.0000000000001, 10000
33  Data 1.0000000000001, 100000
34  Data 1.0000000000001, 1000000
35  Data 1.0000000000001, 10000000
36  Data 0,0
```

実行結果

```
 1  A= 1.0000000000001 N= 1000 N*A= 1000.0000000001
 2  方法１：  1000.00000000009999
 3  方法２：  1000.0000000001
 4
 5  A= 1.0000000000001 N= 10000 N*A= 10000.000000001
 6  方法１：  10000.0000000010002
 7  方法２：  10000.000000001
 8
 9  A= 1.0000000000001 N= 100000 N*A= 100000.00000001
10  方法１：  100000.000000009959
11  方法２：  100000.00000001
12
13  A= 1.0000000000001 N= 1000000 N*A= 1000000.0000001
14  方法１：  1000000.00000010997
15  方法２：  1000000.0000001
16
17  A= 1.0000000000001 N= 10000000 N*A= 10000000.000001
18  方法１：  10000000.0000002347
19  方法２：  10000000.000001
20  OK
```

２．オーバーフロー・アンダーフロー

計算結果の絶対値がコンピュータの処理できる最大の数を越えてしまう現象を**オーバーフロー**という。

オーバーフローの確認。

●プログラム(MA211.bas)

```
1  ' << MA211.bas >>
2  ' オーバーフロー
3  '
4  ' 10**nを計算していき、オーバーフローを確認する。
5  F=1.0E+4930
6  For N=4931 To 4935
7    F=F*10.0
8    Print"[";N;"] ";F
9  Next N
10 End
```

実行結果

```
1  [ 4931]  1E4931
2  [ 4932]  1E4932
3  エラー:演算子 * の計算で結果が扱える範囲を越えました。
4  (5行,  5 桁)
5  OK
```

計算結果の絶対値がコンピュータの処理できる最小数以下になる現象をいう。
アンダーフローが生じると計算結果を０にすることが多い。

アンダーフローの確認。

●プログラム(MA212.bas)

```
1  ' << MA212.bas >>
2  ' アンダーフロー
3  '
4  ' 10**nを計算していき、アンダーフローを確認する。
5  F=1.0E-4929
6  For N=-4930 To -4932 Step -1
7    F=F/10.0
8    Print"[";N;"] ";F
9  Next N
10 End
```

実行結果

```
1  [-4930]  1E-4930
2  [-4931]  1E-4931
3  [-4932]  0
4  OK
```

●アンダーフロー・オーバーフローの回避。

ａのｎ乗(ａ：実数、ｎ：整数)をできるだけ正確に計算する問題を考察する。

2.0 の 1000 乗 ≒ 0.107151×10の302 乗
2.0 の -1000 乗 ≒ 0.933264×10の-301 乗

となるが、aのｎ乗を計算する場合、工夫をしなければ、オーバーフロー・アンダーフローが発生する。

数値を仮数部（0.1以上1未満）と指数部に分けて表現し、乗算の結果、仮数部が1以上または0.1未満になると、0.1以上1未満になるように調整することで、オーバーフロー・アンダーフローを回避できる。

○5の2乗の場合。

初期値。
1.0 ＊ 10 の 0 乗
5を掛ける。
（調整前）5.0 ＊ 10 の 0 乗
（調整後）0.5 ＊ 10 の 1 乗
5を掛ける。
（調整前）2.5 ＊ 10 の 1 乗
（調整後）0.25 ＊ 10 の 2 乗

5の(-2)乗の場合。
まず、5の2乗を求め、その後逆数を計算する。
5の2乗：0.25 ＊ 10 の 2 乗
5の(-2)乗 = (1/0.25) ＊ 10 の (-2)乗
= 4.0 ＊ 10 の (-2)乗
= 0.4 ＊ 10 の (-1)乗

●プログラム(MA213.bas)

```
1  ' << MA213.bas >>
2  ' オーバーフロー、アンダーフローの回避
3  '
4  Do
5    ' AとNの読み込み。
6    Read A,N
7    If ( A = 0 ) or (N = 0) Then Exit Do
8    '
9    M=Abs(N)
10   F=1.0: ' f：仮数部。
11   E=0:   ' e：指数部。
12   For I=1 To M
13     F=F*A
14     While F >= 1: F=F/10.0: E=E+1:Wend
15     While F < 0.1: F=F*10.0: E=E-1: Wend
16   Next I
17   If N < 0 Then
18     F=1.0/F
19     E=-E
20   End If
21   While F >= 1: F=F/10.0: E=E+1: Wend
22   While F < 0.1: F=F*10.0: E=E-1: Wend
23   ' 計算結果の表示。
24   Print A;"の";N;"乗 = ";F;"×10の";E;"乗"
25 Loop
26 End
27 '
28 ' データ。
29 Data 2.0, 1000, 2.0, -1000, 0,0
```

実行結果

```
 2の 10乗 =  0.1024×10の 4乗
 2の 1000乗 =  0.107150860718626732×10の 302乗
 2の-1000乗 =  0.933263618503218879×10の-301乗
OK
```

３．桁落ち

12.34 － 12.33 ＝ 0.01 のように、同符号で絶対値のほぼ等しい２個の数に対して、減算を行うと有効数字が著しく減ってしまう現象をいう。

●桁落ちの発生と回避。

円周率（π＝ 3.14159 26535 89793 …）を半径1の円を内部から近似する正2^n角形の周囲の長さで求めることを考える。

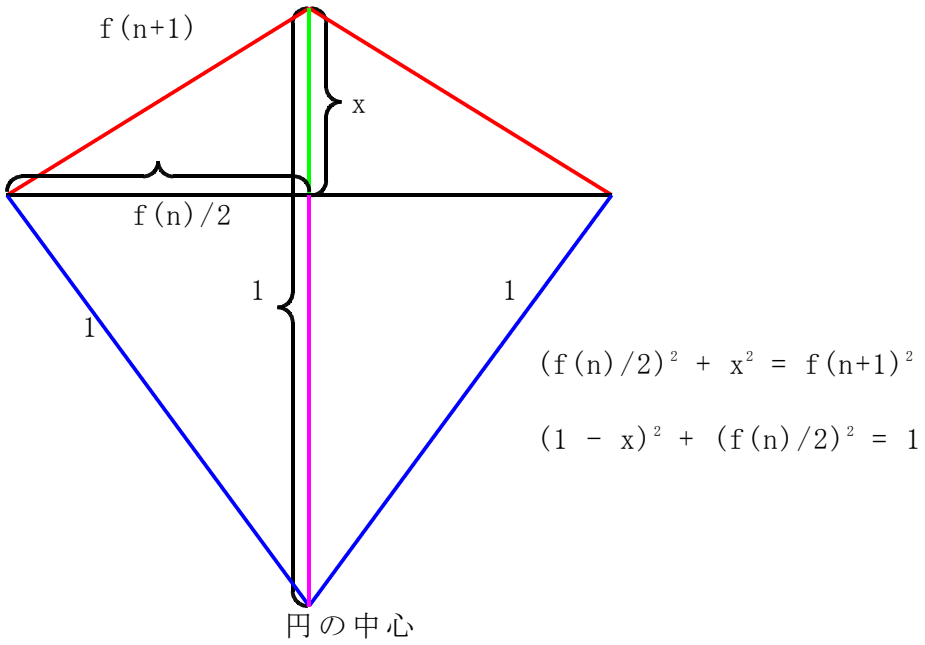
最初、正 2^2 角形で近似し、一辺の長さをf(2)＝ $\sqrt{2}$ とする。

正2^n角形の一辺f(n)と正2^{n+1}角形の一辺f(n+1)の関係は、

$$f(n+1) = \sqrt{2 - \sqrt{4 - f(n)^2}}$$

である。円周率は、$\frac{2^n f(n)}{2}$ となる。

（ヒント）



$(f(n)/2)^2 + x^2 = f(n+1)^2$

$(1 - x)^2 + (f(n)/2)^2 = 1$

●プログラム(MA311a.bas)

桁落ちが発生する。

```
1  ' << MA311a.bas >>
2  ' 桁落ちの発生例
3  '
4  Print"π = ";Pi
5  Print" n                    f(n)     π-近似値"
6  '
```

```
 7 W=4.0: ' w=2^2。
 8 F=Sqr(2.0)
 9 For N=2 To 34
10   F=Sqr(2.0-sqr(4.0-F*F))
11   W=W*2.0: ' w=2^n。
12   Print Using"## ";N;
13   Print Using"##.####################  ";F;
14   Print Using"##.####################";Pi-W*F/2
15 Next N
16 End
```

実行結果

```
π  =  3.14159265358979324
 n                      f(n)     π-近似値
 2  0.76536686473017954000   0.08012519466907506000
 3  0.39018064403225654000   0.02014750133174095000
 4  0.19603428065912120000   0.00504416304385397000
 5  0.09813534865483603000   0.00126149663504032000
 6  0.04908245704582458000   0.00031540265702035000
 7  0.02454307657143985000   0.00007885244549207000
 8  0.01227176929830895000   0.00001971322270106000
 9  0.00613591352593195000   0.00000492831263253000
10  0.00306796037256953000   0.00000123207858986000
11  0.00153398063748541000   0.00000030801968098000
12  0.00076699037514281000   0.00000007700483790000
13  0.00038349519462142000   0.00000001925109360000
14  0.00019174759819194000   0.00000000481309194000
15  0.00009587379920595000   0.00000000120938156000
16  0.00004793689961626000   0.00000000033856209000
17  0.00002396844980870000   0.00000000026444979000
18  0.00001198422490548000  -0.00000000003199939000
19  0.00000599211245500000  -0.00000000121779612000
20  0.00000299605623655000  -0.00000001070416995000
21  0.00000149802812732000  -0.00000002967691752000
22  0.00000074901404557000   0.00000004621407346000
23  0.00000037450705897000  -0.00000025734987946000
24  0.00000018725345711000   0.00000095690610818000
25  0.00000009362687331000  -0.00000390011502645000
26  0.00000004681343665000  -0.00000390011502645000
27  0.00000002340613932000   0.00007381312424543000
28  0.00000001170191157000   0.00038468530752747000
29  0.00000000585327162000  -0.00085861890434044000
30  0.00000000292663581000  -0.00085861890434044000
31  0.00000000147255029000  -0.02068500657858609000
32  0.00000000073627514000  -0.02068500657858609000
33  0.00000000032927225000   0.31316552884360314000
34  0.00000000000000000000   3.14159265358979324000
OK
```

（考察）有理化

$$\frac{a-b}{\sqrt{a}-\sqrt{b}} = \frac{a-b}{\sqrt{a}-\sqrt{b}}\ \frac{\sqrt{a}+\sqrt{b}}{\sqrt{a}+\sqrt{b}} = \sqrt{a}+\sqrt{b}$$

有理化によって、分母の桁落ちを防ぐことができる。

この場合、f(n)は0に近づくので、f(n)からf(n+1)を計算するときに、桁落ちが生じている。桁落ちを防ぐために、f(n)の右辺を有理化する。

g(2) =	$\sqrt{2}$
g(n+1)=	$\frac{g(n)}{\sqrt{2+\sqrt{4-g(n)^2}}}$

円周率は、2^n*g(n)/2　となる。

●プログラム(MA311b.bas)

桁落ちを回避する。

```
1  ' << MA311b.bas >>
2  '
3  Print"π = ";Pi
4  Print" n                    f(n)     π-近似値"
5  '
6  W=4.0: ' w=2^2。
7  G=Sqr(2.0)
8  For N=2 To 36
9    G=G/Sqr(2.0+Sqr(4.0-G*G))
10   W=W*2.0: ' w=2^n。
11   Print Using"## ";N;
12   Print Using"##.####################  ";G;
13   Print Using"##.####################";Pi-W*G/2
14 Next N
15 End
```

実行結果

```
π  =  3.14159265358979324
 n                    f(n)     π-近似値
 2  0.76536686473017954000   0.08012519466907506000
 3  0.39018064403225654000   0.02014750133174095000
 4  0.19603428065912120000   0.00504416304385397000
 5  0.09813534865483603000   0.00126149663504033000
 6  0.04908245704582458000   0.00031540265702037000
 7  0.02454307657143985000   0.00007885244549216000
 8  0.01227176929830895000   0.00001971322270185000
 9  0.00613591352593195000   0.00000492831263354000
10  0.00306796037256953000   0.00000123207859326000
11  0.00153398063748541000   0.00000030801967550000
12  0.00076699037514279000   0.00000007700492057000
13  0.00038349519462141000   0.00000001925123025000
14  0.00019174759819195000   0.00000000481280757000
15  0.00009587379920613000   0.00000000120320189000
16  0.00004793689961684000   0.00000000030080047000
17  0.00002396844981014000   0.00000000007520012000
18  0.00001198422490528000   0.00000000001880003000
19  0.00000599211245267000   0.00000000000470001000
20  0.00000299605622634000   0.00000000000117500000
21  0.00000149802811317000   0.00000000000029375000
22  0.00000074901405658000   0.00000000000007344000
23  0.00000037450702829000   0.00000000000001836000
24  0.00000018725351415000   0.00000000000000459000
25  0.00000009362675707000   0.00000000000000115000
26  0.00000004681337854000   0.00000000000000029000
27  0.00000002340668927000   0.00000000000000007000
28  0.00000001170334463000   0.00000000000000002000
29  0.00000000585167232000   0.00000000000000000000
30  0.00000000292583616000   0.00000000000000000000
31  0.00000000146291808000  -0.00000000000000000000
32  0.00000000073145904000  -0.00000000000000000000
33  0.00000000036572952000  -0.00000000000000000000
34  0.00000000018286476000  -0.00000000000000000000
35  0.00000000009143238000  -0.00000000000000000000
36  0.00000000004571619000  -0.00000000000000000000
OK
```

４．丸め誤差の累積

コンピュータの内部で処理できる数値の桁数が決まっているため、丸め（４捨５入、切り捨て、切り上げ）が起こる。このために生じる誤差を**丸め誤差**という。丸め誤差が累積して異常な結果となる例を示す。

●丸め誤差が発生する漸化式

$$f(n) = f(n-1) + f(n-2) \quad (n>=2),\ f(1)=b,\ f(0)=1$$

$$b = \frac{1-\sqrt{5}}{2} = -0.618034...$$

厳密解は、$f(n)=b^n$ となる。

数値解をg(n)とし、g(1)にbを代入したときに生じた丸め誤差をeとすると、丸め誤差eの累積していく様子はつぎのようになる。

```
g(0) = f(0)       = 1
g(1) = f(1)+e     = b+e
g(2) = g(1)+g(0) = f(1)+e+f(0)      = f(2)+e
g(3) = g(2)+g(1) = f(2)+e+f(1)+e    = f(3)+2e
g(4) = g(3)+g(2) = f(3)+2e+f(2)+e   = f(4)+3e
g(5) = g(4)+g(3) = f(4)+3e+f(3)+2e = f(5)+5e
g(6) = g(5)+g(4) = f(5)+5e+f(4)+3e = f(6)+8e
g(7) = g(6)+g(5) = f(6)+8e+f(5)+5e = f(7)+13e
...
```

●プログラム(MA411.bas)

```
1  ' << MA411.bas >>
2  '
3  Dim F(999),G(999)
4  '
5  N=200: B=(1.0-Sqr(5.0))/2.0
6  Print"                厳密解            数値解"
7  F(0)=1.0: F(1)=b
8  G(0)=1.0: G(1)=b
9  Print"(  0) ";F(0);"  ";G(0)
10 Print"(  1) ";F(1);"  ";G(1)
11 '
12 For I=2 To N
13   F(I)=B*F(I-1)
14   G(I)=G(I-1)+G(I-2)
15   If I Mod 10 = 0 Then
16     Print Using"(###)";I;
17     Print F(I);"  ";G(I)
18   End If
19 Next I
20 End
```

実行結果

```
                厳密解              数値解
(  0)  1    1
(  1) -0.618033988749894848  -0.618033988749894848
( 10) 0.00813061875578334875   0.00813061875578334875
( 20) 6.61069613518959701E-5   6.61069613518782242E-5
( 30) 5.37490499855570335E-7   5.37490497672966916E-7
( 40) 4.37013033918106746E-9   4.36986189670651032E-9
( 50) 3.55318637009634346E-11   2.51562193385318489E-12
( 60) 2.88896037434990854E-13  -4.06044039884256858E-9
( 70) 2.34890354044042508E-15  -4.99436684679569788E-7
( 80) 1.90980391814308318E-17  -6.14266517751882414E-5
( 90) 1.55278875567226794E-19  -0.00755497873166347412
(100) 1.26251333806384294E-21  -0.929200957342832128
(110) 1.02650146258885251E-23  -114.284162774436688
(120) 8.34609204456396368E-26  -14056.0228202983698
(130) 6.78588925150259597E-28  -1728776.52273392505
(140) 5.51734784229356366E-30  -212625456.273452483
(150) 4.48594518487328384E-32  -26151202345.1119215
(160) 3.64735100575467234E-34  -3216385262992.49289
(170) 2.96552204963141997E-36  -395589236145731.514
(180) 2.41115291974223021E-38  -48654259660661983.7
(190) 1.96041651523179603E-40  -5.98407834902527826E18
(200) 1.59393992878910738E-42  -7.35992982670448565E20
OK
```

（考察）

厳密解f(n)と数値解g(n)の違いが顕著になる理由は、丸め誤差eにかかる係数が、フィボナッチ数のように大きくなるからである。