

# 浮動小数点数演算の誤差と有効数字の桁数

ご購入はこちら

高田 昌之

リスト1 小さな値を大量に足す(AddingTiny.c)

10<sup>-n</sup>を10<sup>n</sup>回足すプログラム, nは引数で指定できる

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>

int main ( int argc, char **argv ) {
    float      data4, sum4; /* 32ビット */
    double     data8, sum8; /* 64ビット */
    long double data16, sum16; /* 128ビット */
    int        o;           /* 32ビット */
    long int   n, c;       /* 64ビット */
    char       str[256];

    if ( argc >= 2 ) {
        o = strtol( argv[1], NULL, 10 );
    } else {
        printf( "> " );
        o = strtol( fgets( str, 256, stdin ), NULL, 10 );
    }

    n = 1;
    for( c=0; c<o; c++ ) { /* n回繰り返すときの定石 */
        n *= 10;             /* 10のn乗 */
    }
}

data16 = ((long double) 1.0) / n;
data8  = (double) data16;
data4  = (float) data16;
sum4   = 0;
sum8   = 0;
sum16 = 0;

for( c=0; c<n; c++ ) { /* n回繰り返すときの定石 */
    sum4 += data4;
    sum8 += data8;
    sum16 += data16;
}

printf( "Ans4: %40.8f, Err4: %8.1e\n", sum4, sum4 - 1 );
printf( "Ans8: %40.18lf, Err8: %8.11e\n", sum8, sum8 - 1 );
printf( "Ans?: %40.33Le, Err?: %8.1Le\n", sum16, sum16 - 1 );

return 0;
}
```

リスト2

10<sup>-n</sup>を10<sup>n</sup>回足すプログラムの実行結果

<span style="border: 1px solid black; border-radius: 50%; padding: 2px;">float</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">double</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">long double</span>	takata@Air24:~/FPN[1234] ./AddingTiny 5 Ans4: 1.00099015, Err4: 9.9e-04 Ans8: 0.9999999998083755, Err8: -1.9e-12 Ans?: 9.99999999980837550594969798112288e-01, Err?: -1.9e-12 takata@Air24:~/FPN[1235] ./AddingTiny 6 Ans4: 1.00903893, Err4: 9.0e-03 Ans8: 1.000000000007918111, Err8: 7.9e-12 Ans?: 1.00000000000791811061126616446301e+00, Err?: 7.9e-12 省略 takata@Air24:~/FPN[1239] ./AddingTiny 10 Ans4: 0.00195312, Err4: -1.0e+00 Ans8: 1.00000069475623476, Err8: 6.9e-08 Ans?: 1.00000069475623476478176598902792e+00, Err?: 6.9e-08
--	---

10<sup>-n</sup>を10<sup>n</sup>回足した値

誤差

ます。

## 変数型によって異なる演算精度

仮数のビット幅が小さくて有効数字の桁数を十分取れない計算機で、絶対値の大きな数と小さな数を加減算する場合には、その小さな数が大きな数に与える影響が小さく、演算結果から消えてしまう可能性があり

### ● 小さな数を大量に足した場合

リスト1のプログラムは、10<sup>-n</sup>を10<sup>n</sup>回足した結果を表示します。入力値nを5～10で変え、実行しました(リスト2)。各行の左側の値は10<sup>-n</sup>を10<sup>n</sup>回足した値であり、誤差がなければ1になります。右側はその