誤差の怖い話 — 浮動小数点数

樋口さぶろお

龍谷大学理工学部数理情報学科

数値計算法 L02(2010-04-16)

今日の目標

1. 浮動小数点数はどんな形でメモリに置かれてるか
2. いろんな誤差
3. 人間による有効数字を考慮した計算

http://hig3.net

指数表記または科学表記

物理実験

有効数字 $s+1$ 桁の指数表記

$$\pm \mathrm{n}_0.\mathrm{n}_1\mathrm{n}_2\mathrm{n}_3\mathrm{n}_4\cdots\mathrm{n}_s \times 10^e$$

符号 $\pm$
仮数部 $\mathrm{n}_0.\mathrm{n}_1\mathrm{n}_2\mathrm{n}_3\mathrm{n}_4\cdots\mathrm{n}_s$ ($s+1$ 個の数字)
指数部 e (整数)

例 -1.2345×10^3. **仮数部の 0 桁目は** $1\le \mathrm{n}_0\le 9$ **と限定**

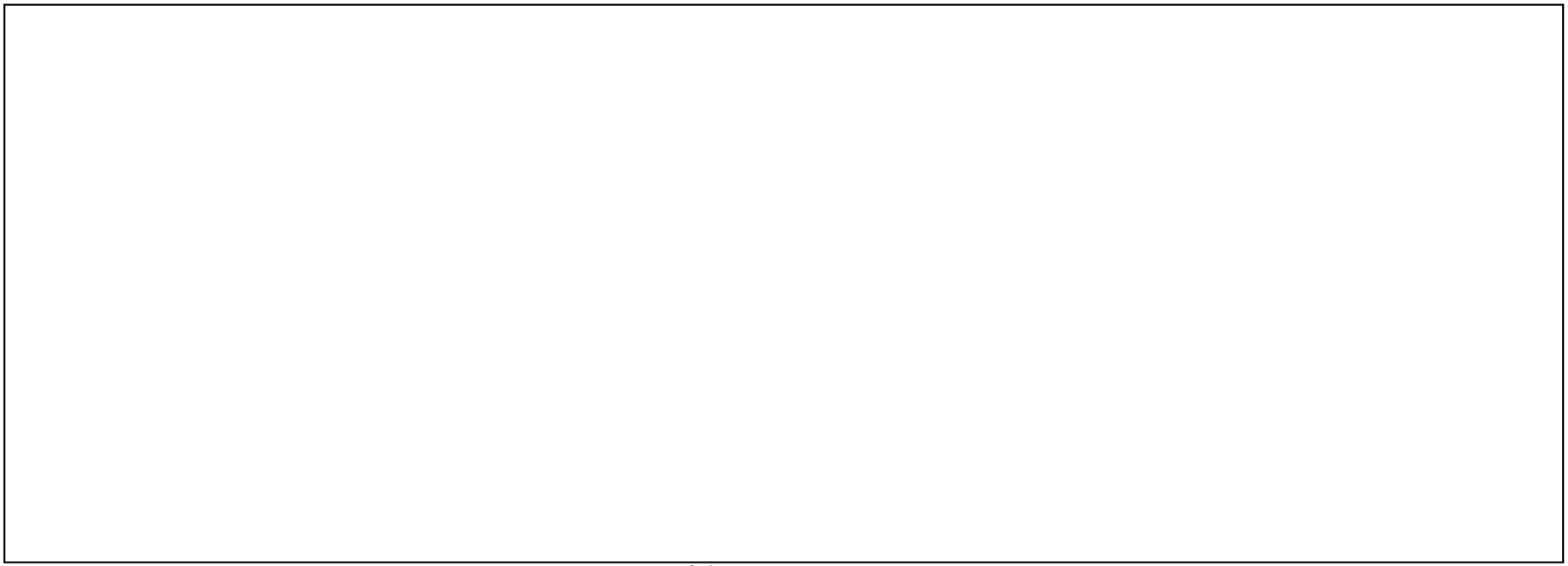
$\mathrm{n}_0.\mathrm{n}_1\mathrm{n}_2\mathrm{n}_3\mathrm{n}_4\cdots\mathrm{n}_s = 1.2345$ **のとき, 真の値** x **は** $1.23445\le x<1.23455$ **というつもり**. 5 **桁分の数字を信じていい**. **有効数字は** 5 **桁**.

- **地球の質量** 5.972×10^{24} kg.
- **電子の質量** 9.10938188×10^{-31} kg
- 1kg **の分銅の質量** 1.000000000×10^{0} kg

指数表記の利点

1. **有効数字を明確にできる.**
   いい分銅 1.0000000×10^0 kg, **安物の分銅** 1.000×10^0kg
2. **エコ**

8 桁電卓の表示領域のエコな使い方



同じ 8 桁でも指数表記なら広い範囲を記述できる.

指数表記で答えよう!

例題

- 富士山の高さは何 *m*? 有効数字 *3* 桁で.
- 日本の人口は何人? 有効数字 *2* 桁で.
- 自分の身長は何 *cm*? 自分の知ってるだけの有効数字で.

浮動小数点数って指数表記的なメモリ格納方法

栗原 §1.2

`double` は 64bit, `float` は 32bit の **浮動小数点数**.

指数表記を使って代入することができる.

E□ $= \times 10^{\square}$

代入

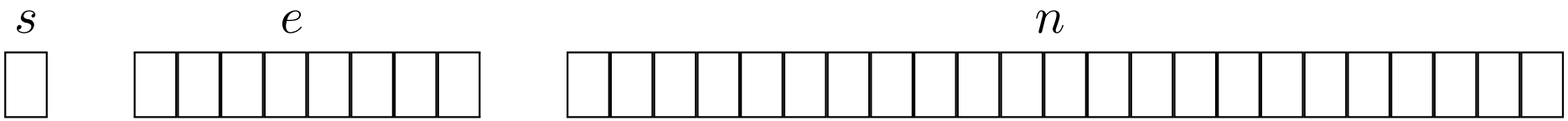
```
double x=838.88; /*x = 8.3888 × 10^2*/
double y=-1.2345E-12; /*指数表記 y = −1.2345 × 10^−12*/
float z=3.879E+9; /*指数表記 z = 3.879 × 10^9*/
```

浮動 (floating) 小数点 (point)って?

-0.000000000012345 や 3879000000.0 なのに, 指数部を使って便利な場所に小数点を動かしてるでしょ.

浮動小数点数のデータ構造

float(32bit) の場合



符号 s 1bit $s = \pm 1$. $0 \to s = +1, 1 \to s = -1$.

指数部 e 8bit $0 \le e \le 255$.

仮数部 n 23 bit $0 \le n \le 2^{23} - 1$.

表す数: $s \times (1 + 2^{-24} n) \times 2^{e-127}$.

あれっなんかすごく複雑になってる…要するに.

やっぱり字幕 (2 進) より吹替 (10 進) が楽

ここからは

なんちゃって 10 進浮動小数点数 myfloat のおとぎ話

または

指数表記による有効数字を考慮した手計算のたとえ話

で中継します.

すべての数は

$$\pm \mathrm{n}_0.\mathrm{n}_1\mathrm{n}_2\mathrm{n}_3\mathrm{n}_4\mathrm{n}_5 \times 10^{\pm \mathrm{e}_1\mathrm{e}_2}$$

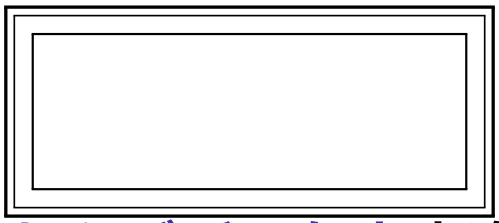
適宜頭の中で翻訳してお楽しみください.

代入で大ピ〜ンチ!

栗原 §1.2

```
myfloat x=1.234567890E9;
```

1.23457×10^9 が代入されて少しずれる.

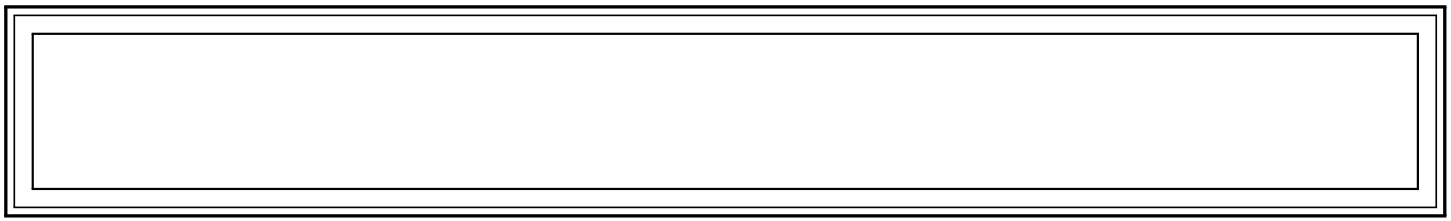


おとぎ話の欠点実際は 10 進でなく 2 進で記録される.
3.140000000000 のような, 10 進であるところから先が 0 の数にも丸め誤差がでる.
⇝ きょうの quiz

かけ算で大ピ〜ンチ!

$$1.23456 \times 10^{13} \times 7.89012 \times 10^{-42} = 9.740826\cancel{6547} \times 10^{-29}$$

積の有効数字の桁数は 5 桁のまま変わらない.
~~消したところは~~, 有効数字以下の影響を受けるので信じられない.
だって, 有効数字 1 桁のときを想像してみて.
$2 \times 10^2 \times 3 \times 10^5 = 6.0 \times 10^7$ **だけど**, $3 \to 3.1$ **で** $6 \to 6.2$ **でしょ**.



- $5.00000 \times 10^{40}.00000 \times 10^{59} = 1.5000 \times 10^{100}$ **は** ∞ **として扱われる**.
- $5.00000 \times 10^{-40}.00000 \times 10^{-61} = 1.5000 \times 10^{-100}$ 0 **として扱われる**.

オーバーフロー, アンダーフローは加算減算でも起きる (もちろん).
手計算の場合, 指数部の桁数には制限をつけないのでこの問題は起きない

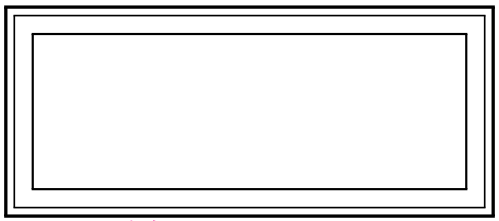
足し算引き算で大ピ〜ンチ!

栗原 §1.2

$$1.23456 \times 10^4 + 7.89012 \times 10^0 = ??$$

$$\begin{array}{rrl} & 12345. & 6 \\ +) & 7. & 89012 \\ \hline & 12353. & 4\cancel{9012} \rightarrow 1.23534 \times 10^4. \end{array}$$

有効数字は 6 桁のまま.
下の方の桁 9012 の情報が使われない.



手計算の場合, 有効数字よりも 1 桁多くとって計算して最後に四捨五入するのが普通

足し算引き算でもっとひどい大ピ〜ンチ!

$$1.23456 \times 10^4 + 7.89012 \times 10^{-3} = ??$$

$$\begin{array}{rr} 12345. & 6 \\ +) \quad 0. & 00789012 \\ \hline 12345. & 6\cancel{0789012} \rightarrow 1.23456 \times 10^4. \end{array}$$

2 個目の数の情報がまったく消えてしまう. 数を足したのに変化しない.

危険なプログラム

```
int i;
myfloat x=1;
for(i=0;i<1.0E+7;i++){
    x=x+1.0E-7;
}
```

最後に x **は** 2 **になるか?**

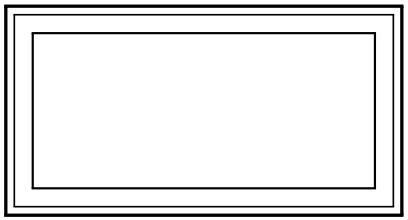
float double の数を多くの回数加えるのは危険

引き算の大ピ〜ンチ!

栗原 §1.2

$$1.23456 \times 10^5 - 1.23442 \times 10^5 = 1.2 \times 10^2$$

有効数字 2 桁に減っちゃう!
近い数同士の差を取ると有効数字が減って精度が落ちる.



演算順序で大ピ〜ンチ!!

```
myfloat x=1.0+1E-10-1.0;/*=(1.0+1E-10)-1.0;*/
myfloat y=1e-10;

myfloat z=1.0E-51 * 1.0E-51 * 1.0E+51;
myfloat w=1.0E-51 * 1.0E+51 * 1.0E-51;

myfloat a=(1.00000E5+1.23456E0)*(1.00000E5+1.23456E0)
-1.00000E5*1.00000E5; /*(c+d)(c+d)-c^2*/
myfloat b=2.0*1.23456E0+1.23456E0*1.23456E0; /*2cd+d^2*/
```

結果は同じか? 栗原 例題 14,15

例題

仮数部の有効数字 *3* 桁の *myfloat* を使って, 次の計算結果を求めよう.

- *1.00E2+1.23E0*
- *(1.00E2+1.23E0)*(1.00E2+1.23E0)-1.00E2*1.00E2*

今日出てきた誤差の種類

- オーバーフロー
- アンダーフロー
- 丸め誤差
- 情報落ち
- 桁落ち

今後出てくる誤差

- 打ち切り誤差
- 離散化誤差