コンピュータサイエンス第２

南出 靖彦

第６回：前半

最近接偶数への丸め

- 原則は 0 捨 1 入。
- ただし 53 桁目以降が $1000\cdots$（先頭だけ 1 で残りは全部 0）のとき
    - 52 桁目（仮数部の一番下）が 0 なら切り捨て，1 なら切り上げ。

前回課題４ (a) について

以下は，仮数部分（先頭の 1 を含めて 53 ビット）を枠で囲んだ 2 進数である。

$$
\begin{aligned}
2^{52}-1 &= \boxed{11\cdots1.0} \\
2^{52} &= \boxed{10\cdots00}. \\
2^{52}+1 &= \boxed{10\cdots01}. \\
2^{52}+1.5 &= \boxed{10\cdots01}.1 \\
&\overset{\text{丸め}}{\to} \boxed{10\cdots10}.
\end{aligned}
$$

$$
\begin{aligned}
2^{53}-1 &= \boxed{11\cdots11}. \\
2^{53} &= \boxed{10\cdots00}0 \\
2^{53}+1 &= \boxed{10\cdots00}1. \\
&\overset{\text{丸め}}{\to} \boxed{10\cdots00}0. \\
2^{53}+1.5 &= \boxed{10\cdots00}1.1 \\
&\overset{\text{丸め}}{\to} \boxed{10\cdots01}0.
\end{aligned}
$$

$$
\begin{aligned}
2^{54} - 1 &= \quad \boxed{11\cdots 11}1. \\
&\xrightarrow{\text{丸め}} \boxed{10\cdots 00}00. \\
2^{54} &= \boxed{10\cdots 00}00. \\
2^{54} + 1 &= \boxed{10\cdots 00}01. \\
&\xrightarrow{\text{丸め}} \boxed{10\cdots 00}00. \\
2^{54} + 1.5 &= \boxed{10\cdots 00}01.1 \\
&\xrightarrow{\text{丸め}} \boxed{10\cdots 00}00.
\end{aligned}
$$

指数部の値（小数点の位置）が大きく異なるものどうしの足し算をすると，小さい方の値の情報が少なくなったり，無くなったりしてしまう。

「情報落ち」と呼ばれる現象である。

前回課題４ (c) について

- プログラム３

$$g(n) = (\cdots(((1 + \frac{1}{2}) + \frac{1}{4}) + \frac{1}{8}) + \cdots + \frac{1}{2^n})$$

- e_n

$$e_n = (\cdots(((1 + \frac{1}{1}) + \frac{1}{1 \cdot 2}) + \frac{1}{1 \cdot 2 \cdot 3}) + \cdots + \frac{1}{n!})$$

- e'_n

$$e'_n = (\cdots(((\frac{1}{n!} + \frac{1}{(n-1)!}) + \frac{1}{(n-2)!}) + \cdots + 1))$$

e_n と e'_n を計算するプログラム

- e_n：プログラム３のサブルーチン `t` の定義中の `a = a * 2` を `a = a * i` に変更
- e'_n: さらにサブルーチン `g` の定義中の `a = a + t(i)` を `a = a + t(n-i)` に変更

これらの計算は共に $n = 17$ で収束する。

- 収束した値 e'_{17} は定数 `math.e` に等しいが e_{17} の値は異なる。
- これは大きい数から足し算をすると，足し算をするふたつの数の値の差がどんどん開いていき情報落ちが多く発生するからである。

足し算の順番が結果に影響する例を説明する。

- 仮数は先頭の 1 を含めて 6 ビットとする。
- 仮数に下線を引く。

大きい値から足し算すると切り上げが多く起こりすぎて値が大きくなってしまう例

$$
\begin{aligned}
a &= 0.\underline{100000} \\
b &= 0.000\underline{100101} \\
c &= 0.000000\underline{100001} \\
\text{総和} &= 0.\underline{100101}001001 \\
&\xrightarrow{\text{丸め}} 0.\underline{100101}\text{(切り捨て)} \\
&\quad\ \text{これが仮数 6 桁での最良近似}
\end{aligned}
$$

$$
\begin{aligned}
a &= 0.\underline{100000} \\
b &= 0.000\underline{100101} \\
c &= 0.000000\underline{100001} \\
& \quad 0.\underline{100101}\text{(切り捨て)} \\
& \quad \text{これが仮数 6 桁での最良近似}
\end{aligned}
$$

$$
\begin{aligned}
c + b &= 0.000\underline{101001}001 \\
&\xrightarrow{\text{丸め}} 0.000\underline{101001}\text{(切り捨て)} \\
\text{これに } a \text{ を足す} &= 0.\underline{100101}001 \\
&\xrightarrow{\text{丸め}} 0.\underline{100101}\text{(切り捨て)} \\
& \quad \text{上の値に等しい}
\end{aligned}
$$

$$
\begin{aligned}
a + b &= 0.\underline{100100}101 \\
&\xrightarrow{\text{丸め}} 0.\underline{100101}\text{(切り上げ)} \\
\text{これに } c \text{ を足す} &= 0.\underline{100101}100001 \\
&\xrightarrow{\text{丸め}} 0.\underline{100110}\text{(切り上げ)} \\
& \quad \text{上の値より大きい}
\end{aligned}
$$

大きい値から足し算すると切り捨てが多く起こりすぎて値が小さくなってしまう例

$$
\begin{aligned}
x &= 0.\underline{100000} \\
y &= 0.000\underline{100010} \\
z &= 0.0000000\underline{110000} \\
\text{総和} &= 0.\underline{100100}1010000 \\
&\overset{\text{丸め}}{\longrightarrow} 0.\underline{100101}(\text{切り上げ}) \\
&\ \text{これが仮数 6 桁での最良近似} \\
\\
z + y &= 0.000\underline{100101}0000 \\
&\overset{\text{丸め}}{\longrightarrow} 0.000\underline{100101}(\text{切り捨て}) \\
\text{これに } x \text{ を足す} &= 0.\underline{100100}101 \\
&\overset{\text{丸め}}{\longrightarrow} 0.\underline{100101}(\text{切り上げ}) \\
&\ \text{上の値に等しい} \\
\\
x + y &= 0.\underline{100100}010 \\
&\overset{\text{丸め}}{\longrightarrow} 0.\underline{100100}(\text{切り捨て}) \\
\text{これに } z \text{ を足す} &= 0.\underline{100100}0110000 \\
&\overset{\text{丸め}}{\longrightarrow} 0.\underline{100100}(\text{切り捨て}) \\
&\ \text{上の値より小さい}
\end{aligned}
$$

前回課題4 (b) について

`math.e` の値

$\underline{2.718281828459045}0907955982984276488423347473144531250000000000\cdots$

下線部分が正しい e の値に一致する。

- これが 53 ビット（先頭の 1 を含めて）の仮数で表せるネイピア数の最良の近似である。

5 ビット（先頭の 1 を含めて）の仮数で円周率 $\pi = 3.141592\cdots$ を表す場合，

二進数：$11.001 =$ 十進数：$\underline{3.1}250000\cdots$（下線部分が正しい π の値に一致）

が最良の近似である。

まとめ

固定ビット数の２進数で実数を扱っている限り，必ず誤差が生じる。
どんな原因でどんな誤差が生じるかを理解していることが必要である。