表現誤差

実際の値は数直線上のどんなに狭い部分にも無限個の実数が含まれているため浮動小数点表示には表現の誤差が含まれる.

*

切り捨て浮動小数点での表示を m 桁までできたとする. そのときそれより先の $m+1$ 以上の桁を切り捨てて表示したとするとその切り捨てた分が誤差になる. 今, 実際の値を z, 浮動小数点で表示できる部分を F, 切り捨てたられた表現誤差を δ_1 とすると,

$$\delta_1 = z - F$$

となる.z と F を浮動小数点で表示すると,

$$\delta_1 = (0.f_1f_2\cdots)_\beta \times (\beta)_{10}{}^{(E)_{10}} - (0.f_1f_2\cdots f_m)_\beta \times (\beta)_{10}{}^{(E)_{10}}$$

となる. ここで省略のため仮数部も指数部も正の値で考えている. (**??**) 式から

$$\begin{aligned}\delta_1 &= \Big((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m)_{10}(\beta)_{10}{}^{-m} + (f_{m+1})_{10}(\beta)_{10}{}^{-(m+1)} + \cdots\Big) \times (\beta)_{10}{}^{(E)_{10}} \\ &\quad - \big((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m)_{10}(\beta)_{10}{}^{-m}\big) \times (\beta)_{10}{}^{(E)_{10}} \\ &= (f_{m+1})_{10}(\beta)_{10}{}^{-(m+1)} \times (\beta)_{10}{}^{(E)_{10}} + (f_{m+2})_{10}(\beta)_{10}{}^{-(m+2)} \times (\beta)_{10}{}^{(E)_{10}} + \cdots .\end{aligned}$$

仮に m が十分大きく $(\beta)_{10}{}^{-(m+2)}$ 以降の項を極小だとする. そのとき, 誤差の値が $m+1$ 桁の値で決まるとみなすと,

$$\delta_1 \approx (f_{m+1})_{10}(\beta)_{10}{}^{-(m+1)} \times (\beta)_{10}{}^{(E)_{10}}.$$

δ_1 として取り得る最大の値を取るのは $(f_{m+1})_{10}$ が最大値を取ったときである. よって, $(f_{m+1})_{10} = (\beta)_{10} - 1$. このとき

$$\begin{aligned}\delta_1 &\le ((\beta)_{10} - 1)(\beta)_{10}{}^{-(m+1)} \times (\beta)_{10}{}^{(E)_{10}} \\ &= \Big((\beta)_{10}{}^{-m} - (\beta)_{10}{}^{-(m+1)}\Big) \times (\beta)_{10}{}^{(E)_{10}}.\end{aligned}$$

δ_1 の相対誤差は δ_1 を F で割ることで求められるので相対誤差を δ_{1r} とすると

$$\begin{aligned}\delta_{1r} &= \frac{\Big((\beta)_{10}{}^{-m} - (\beta)_{10}{}^{-(m+1)}\Big) \times (\beta)_{10}{}^{(E)_{10}}}{\big((f_1)_{10}(\beta)_{10}{}^{-1}\cdots + (f_m)_{10}\beta_{10}{}^{-m}\big) \times (\beta)_{10}{}^{(E)_{10}}} \\ &\approx \frac{(\beta)_{10}{}^{-m}}{(f_1)_{10}(\beta)_{10}{}^{-1}}.\end{aligned} \tag{1.1}$$

相対誤差の取り得る最大の値は $(f_1)_{10} = (1)_{10}$ のときである. よって,

$$\begin{aligned}\delta_{1r} &\leq \frac{(\beta)_{10}{}^{-m}}{(\beta)_{10}{}^{-1}} \\ &= (\beta)_{10}{}^{-(m-1)}\end{aligned} \tag{1.2}$$

となる.

*

四捨五入浮動小数点の形で正確に表せる数の中間に境目を置いて, β 進法で四捨五入のようなことをする場合を考える[1)]. 今, $m+1$ 桁目の値を四捨五入することを考える. このとき四捨五入される $m+1$ 桁目の値を δ_{ro} とすると,

$$\begin{aligned}\delta_{ro} &\equiv (f_{m+1})_{10}(\beta)_{10}{}^{-(m+1)} \times (\beta)_{10}{}^{(E)_{10}} + \cdots \\ &= \begin{cases} (\beta)_{10}{}^{-m} & \left((f_{m+1})_{10} \geq \frac{(\beta)_{10}}{2}\right) \\ 0 & \left((f_{m+1})_{10} \leq \frac{(\beta)_{10}}{2} - 1\right) \end{cases}\end{aligned}$$

である. ここで四捨五入による表現誤差を δ_2 とすると

$$\delta_2 = z - (F + \delta_{ro})$$

となる. 切り捨ての時と同様に浮動小数点で表したとすると,

$$\begin{aligned}\delta_2 &= (0.f_1f_2\cdots)_\beta \times (\beta)_{10}{}^{(E)_{10}} - ((0.f_1f_2\cdots f_m)_\beta \times (\beta)_{10}{}^{(E)_{10}} + \delta_{ro} \\ &= \Big((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m)_{10}(\beta)_{10}{}^{-m} + (f_{m+1})_{10}(\beta)_{10}{}^{-(m+1)} + \cdots\Big) \times (\beta)_{10}{}^{(E)_{10}} \\ &\quad - \left((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m)_{10}(\beta)_{10}{}^{-m}\right) \times (\beta)_{10}{}^{(E)_{10}} - \delta_{ro}.\end{aligned}$$

今 $(f_{m+1})_{10} = \dfrac{(\beta)_{10}}{2}$ とする. このとき δ_{ro} は繰り上がって,

$$\begin{aligned}\delta_2 &= \left((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m)_{10}(\beta)_{10}{}^{-m} + (\frac{(\beta)_{10}}{2}(\beta)_{10}{}^{-(m+1)} + \cdots\right) \times (\beta)_{10}{}^{(E)_{10}} \\ &\quad - \left((f_1)_{10}(\beta)_{10}{}^{-1} + \cdots + (f_m+1)_{10}(\beta)_{10}{}^{-m}\right) \times (\beta)_{10}{}^{(E)_{10}} \\ &= \left(\frac{(\beta)_{10}}{2} - 1)(\beta)_{10}{}^{-m}\right) \times (\beta)_{10}{}^{(E)_{10}} + \cdots \\ &= -\frac{(\beta)_{10}}{2}(\beta)_{10}{}^{-m} \times (\beta)_{10}{}^{(E)_{10}} + \cdots\end{aligned}$$

[1)] ここでの四捨五入とは中間に境目をおいて値がその境目以上のときは繰り上げ, それより低いときは切り捨てを行う丸めのこと.

となる. 切り捨てのときと同様に ${(\beta)_{10}}^{-(m+2)}$ 以降の項が無視できるとすると,

$$\delta = \frac{{\beta_{10}}^{-m}}{2} \times {\beta_{10}}^{E_{10}}$$

となる. $(f_{m+1})_{10}$ の値が $\frac{(\beta)_{10}}{2} - 1$ 以下の時は繰り上がらずそのときの $(f_{m+1})_{10}{(\beta)_{10}}^{-(m+1)}$ が表現誤差になるので最終的に誤差の値が最大になるのは $f_{m+1} = \frac{\beta}{2}$ のときである. 打切り誤差の時と同様に相対誤差 δ_{2r} も求めると

$$\begin{aligned}\delta_{2r} &= \frac{\frac{{\beta_{10}}^{-m}}{2} \times {\beta_{10}}^{E_{10}}}{(f_1)_\beta {\beta_{10}}^{-1} \cdots + (f_m)_\beta {\beta_{10}}^{-m}) \times {\beta_{10}}^{E_{10}}} \\ &\approx \frac{\frac{{\beta_{10}}^{-m}}{2}}{(f_1)_\beta {\beta_{10}}^{-1}} \end{aligned} \tag{1.3}$$

となる. 相対誤差の取り得る最大の値は $(f_1)_{10} = (1)_{10}$ のときである. よって,

$$\begin{aligned}\delta_{2r} &\leq \frac{\frac{{\beta_{10}}^{-m}}{2}}{{\beta_{10}}^{-1}} \\ &= \frac{\beta_{10}^{-(m-1)}}{2}. \end{aligned} \tag{1.4}$$

*

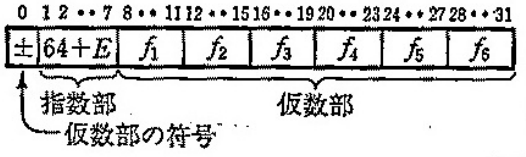
具体例パソコンで使う場合は 2 進数かあるいは 16 進数を用いる[2]. 1 つの数は定まったビット数の 1 語に収められることになっているため[3],1 語で表現しうる数の種類はこのビット数に応じて高々2^{32} 個とか 2^{64} 個とか言うように限られたものになる. そのときには表現誤差が含まれた形になる. 以下に 32 ビット語の場合の代表的な例を挙げる.

*

IBM 方式 IBM 方式とは図**??**の概念図のような形で数が表現されている表現方式である.

IBM 方式は (**??**) 式において $\beta = 16$,$m = 6$ とし, 丸め[4]を切り捨て方式にしている. この形で表現できる数は, 絶対値で約 $16^{-64} \sim 16^{63}$ 範囲である[5]. 10 進表示

[2] 16 進法は 2 進法の 4 桁を一つにまとめたものなので実質は 2 進法である.
[3] ビットはコンピュータの最小単位で 2 進法の 1 桁のこと.
[4] 丸めとは切り捨てなどの端数処理のこと.
[5] 16^{-64} は 0 と表現している.



○ $f_i \neq 0$; 各 f_i は 4 ビットで 0～F のいずれか（章末の Smile (☺) 1 参照）
○指数部 7 ビットを用いて 0～127 を表せるが，これを $E=-64 \sim +63$ に対応させる．
○“0”はこの表現には馴染まない異質な数である．実際には，たとえば，$64+E=0$（すなわち $E=-64$）をそれに当てる．

図 1–1　数の内部表現の概念図（IBM 方式）

図 1.1: IBM 方式の数の内部表現の概念図 (伊理正夫, 1985:数値計算の常識より)

にすると約 $0.86 \times 10^{-77} \sim 0.72 \times 10^{76}$ である．この方式での表現の相対誤差は $(f_1)_{16} = \cdots = (f_6)_{16} = (F)_{16}$ のとき最も小さくなる[6]. (**??**) 式より

$$\begin{aligned}\delta_r &\approx \frac{16^{-6}}{15 \cdot 16^{-1}} \\ &= 16^{-6} \\ &\approx 6 \times 10^{-8}\end{aligned}$$

となる．また,$(f_1)_{16} = (1)_{16}$,$(f_2)_{16} = \cdots = (f_6)_{16} = 0$ のとき最も大きくなる．同様にやると,

$$\begin{aligned}\delta_r &\approx \frac{16^{-6}}{1 \cdot 16^{-1}} \\ &= 16^{-5} \\ &\approx 10^{-6}\end{aligned}$$

となる．

*

IEEE 方式 (マイクロソフト社製 BASIC 等) マイクロソフト社製 BASIC 等の方式は IEEE 方式と呼ばれる表現方式である．この方式は図**??**の概念図のような形で数が表現される．

IEEE 方式は (**??**) 式において $\beta = 2$,$m = 24$ とし, 丸めを四捨五入 (2 進法なので 0 捨 1 入) とする．2 進法で表現されたことで $(f_1)_\beta \neq 0$ の条件から $(f_1)_2$ は自動的

[6] 16 進法では慣習で $0, 1, \cdots, 9$ の他に $10, 11, 12, 13, 14, 15$ に相当するもとのとして A,B,C,D,E,F を使う．

○f_1=1 は明示せず；各 f_i は 0 または 1．
○指数部 8 ビットを用いて 0～255 を表せるが，これを E=−128～127 に対応させる．
○仮数部の符号ビットは “+” のとき 0，“−” のとき 1；符号が “−” のときは仮数部は “補数” 表示とすることもある（ここでの話には関係ないが）．
○“0” はこの表現には馴染まない異質な数である．実際には，たとえば，128+E=0（すなわち E=−128）をそれに当てる．

図 1-2　数の内部表現の概念図（マイクロソフト社製 BASIC 等）

図 1.2: IEEE 方式 (マイクロソフト社製 BASIC 等) の数の内部表現の概念図 (伊理正夫, 1985:数値計算の常識より)

に $(1)_2$ に決まる. そのため $(f_1)_2$ には情報がないことになり省略できるなどの利点がある. この形で表現できる数は, 絶対値で約 $2^{-128} \sim 2^{127}$ の範囲である[7]. 10 進表示にすると約 $2.9 \times 10^{-39} \sim 1.7 \times 10^{38}$ となる. この方式での表現の相対誤差は $(f_1 =)(f_2)_2 = \cdots = (f_{24})_2 = (1)_2$ のとき最小になる. (**??**) 式より

$$\begin{aligned}\delta_r &\approx \frac{\frac{2^{-24}}{2}}{\frac{2}{2}} \\ &= 2^{-25} \\ &\approx 3 \times 10^{-8}.\end{aligned}$$

最大になるのは $(f_1)_2 = (1)_2$,$(f_2)_2 = \cdots = (f_{24})_2 = 0$ のときで最小の時と同様に求めると

$$\begin{aligned}\delta_r &\approx \frac{\frac{2^{-24}}{2}}{\frac{1}{2}} \\ &= 2^{-24} \\ &\approx 6 \times 10^{-8}\end{aligned}$$

である. 表現の相対誤差がほぼ一定であるのが 16 進法に比べて著しい長所の一つである.

[7] 2^{-128} は 0 を表現している.