GFD ワーク 第 1 章は伊理正夫・藤野和建著「数値計算の常識」(以下, 伊理テキスト) の第 1 章を主に参考にしている.

誤差の定義と蓄積

1.1 誤差の定義

量 x の測定値 a に見込まれる誤差が $\Delta a\ (>0)$ であるというときには,

$$x \in (a - \Delta a, a + \Delta a) \tag{1.1}$$

すなわち,

$$a - \Delta a < x < a + \Delta a$$

であることを意味し,

$$x = a \pm \Delta a \tag{1.2}$$

と表記する[1]. 実数 x, y の関数として計算される量 $z = f(x, y)$ を考える. y, z の測定値をそれぞれ b, c, 絶対誤差をそれぞれ $\Delta b, \Delta c$ とすると,

$$z = c \pm \Delta c \tag{1.3}$$

と表せる. ここで,

$$c = f(a, b) \tag{1.4}$$

[1]開区間と閉区間の表記法は以下のとおりである.

開区間 $(a, b) = \{x | a < x < b\}$

閉区間 $[a, b] = \{x | a \leq x \leq b\}$

左閉右開区間, 左閉半開区間 $[a, b) = \{x | a \leq x < b\}$

左開右閉区間, 右閉半開区間 $(a, b] = \{x | a < x \leq b\}$

であり,

$$
\begin{aligned}
(\text{誤差の大きさ}) &= |f(a \pm \Delta a, b \pm \Delta b) - f(a,b)| \\
&= \left| \left(f(a,b) \pm \left(\frac{\partial f(x,y)}{\partial x} \right)_{x=a,y=b} \Delta a \pm \left(\frac{\partial f(x,y)}{\partial y} \right)_{x=a,y=b} \Delta b \pm \cdots \right) - f(a,b) \right| \\
&= \left| \pm \left(\frac{\partial f(x,y)}{\partial x} \right)_{x=a,y=b} \Delta a \pm \left(\frac{\partial f(x,y)}{\partial y} \right)_{x=a,y=b} \Delta b \pm \cdots \right| \\
&\leq \left| \left(\frac{\partial f(x,y)}{\partial x} \right)_{x=a,y=b} \right| \Delta a + \left| \left(\frac{\partial f(x,y)}{\partial y} \right)_{x=a,y=b} \right| \Delta b \\
&\equiv \Delta c
\end{aligned} \tag{1.5}
$$

と表すことができる. ただし, $\Delta a, \Delta b, \Delta c$ はあまり大きくないと想定している.

誤差の蓄積

足し算と掛け算では誤差の蓄積の仕方が異なる. $z = x \pm y$ のとき,

$$
\begin{aligned}
\Delta c &= \left| \left(\frac{\partial f(x,y)}{\partial x} \right)_{x=a,y=b} \right| \Delta a + \left| \left(\frac{\partial f(x,y)}{\partial y} \right)_{x=a,y=b} \right| \Delta b \\
&= \Delta a + \Delta b
\end{aligned} \tag{1.6}
$$

となる. つまり, 絶対誤差 Δc は x, y の絶対誤差 $\Delta a, \Delta b$ の和となる. また, $z = xy$ のとき,

$$
\begin{aligned}
\Delta c &= \left| \left(\frac{\partial f(x,y)}{\partial x} \right)_{x=a,y=b} \right| \Delta a + \left| \left(\frac{\partial f(x,y)}{\partial y} \right)_{x=a,y=b} \right| \Delta b \\
&= b\Delta a + a\Delta b
\end{aligned}
$$

すなわち,

$$
\frac{\Delta c}{c} = \frac{\Delta a}{a} + \frac{\Delta b}{b} \tag{1.7}
$$

となり, z の相対誤差 $\dfrac{\Delta c}{c}$ は x, y の相対誤差 $\dfrac{\Delta a}{a}$, $\dfrac{\Delta b}{b}$ の和となっていることがわかる.

実数の浮動小数点表現とその誤差

実際, 我々は計算機を用いて関数 $f(x,y)$ を計算し, 興味のある値 z を獲得する. よって, Δc には $f(x,y)$ の計算の過程で生じた「計算誤差」が含まれている. 今日, 計算機の性能が向上したからといって, この計算誤差が無視できるほどに小さくできるようになったわけではない. 計算誤差の例を以下に 3 つほど挙げる. 各自実際に計算し, 結果を考察されたい.

例 1

$n = 1, 2, \cdots, 10$ に対して $n^2, n^3, 1/n$ を求め, 表を出力する. 実際に手元の計算機で計算すると, 以下の表のようになる. プログラム例は ex1_ d.f90 を参照されたい.

```
  n         n^2           n^3          1/n
  1     1.00000000    1.00000000    1.00000000
  2     4.00000000    8.00000000    0.50000000
  3     9.00000000   27.00000000    0.33333333
  4    16.00000000   64.00000000    0.25000000
  5    25.00000000  125.00000000    0.20000000
  6    36.00000000  216.00000000    0.16666667
  7    49.00000000  343.00000000    0.14285714
  8    64.00000000  512.00000000    0.12500000
  9    81.00000000  729.00000000    0.11111111
 10   100.00000000 1000.00000000    0.10000000
```

表を見ると, 特におかしなことは起こっていない. しかし, 伊理テキストによると, 同様の計算を大型機の FORTRAN で行うと, $1/10 = 0.1$ とならずに $1/10 = 0.09999996$ となったという[2)].

[2)] 伊理テキスト p.2 に出てくる FORTRAN の FORMAT, F14.8 は, 出力する値を 14 桁分の欄に小

例 2

0.01 を 10,000 回足すプログラムを作り, 結果を出力する. 伊理テキストによると, パソコン BASIC と大型機 FORTRAN では,

$$\sum_{n=1}^{10000} 0.01 = \begin{cases} 100.003 & (\text{パソコン BASIC}) \\ 99.95277 & (\text{大型機 FORTRAN}) \end{cases} \tag{1.8}$$

という結果になったという. 実際に手元の計算機[3] で計算すると,

$$\sum_{n=1}^{10000} 0.01 = \begin{cases} 100.00295 & (\text{単精度}) \\ 100.00000000001425 & (\text{倍精度}) \end{cases} \tag{1.9}$$

と言う結果になる. プログラム例は単精度, 倍精度それぞれ ex2_e.f90, ex2_d.f90 を参照されたい.

例 3

$x = 0.0, 0.1, \cdots, 0.9, 1.0$ に対して x^2 を計算し合計するプログラムを, 伊理テキストのように,

$$\begin{aligned} & x = 0.0\ ;\ s = 0.0\ ; \\ & \text{while } x \le 1.0 \text{ do} \\ & \text{begin } s = s + x^2\ ;\ x = x + 0.1 \text{ end}\ ; \\ & \text{print } s \end{aligned}$$

というような構造で作成する. 用いて作成する. 伊理テキストによると,

$$\sum_{n=0}^{10} (0.1n)^2 = \begin{cases} 2.85 & (\text{パソコン BASIC}) \\ 3.85 & (\text{大型機 FORTRAN}) \end{cases} \tag{1.10}$$

という値になったという. 手元の計算機では,

$$\sum_{n=0}^{10} (0.1n)^2 = \begin{cases} 2.8500004 & (\text{単精度}) \\ 3.8499999999999996 & (\text{倍精度}) \end{cases} \tag{1.11}$$

数点以下 8 桁の精度で書くという意味である. この 14 桁分の欄には符号, 小数点を含むことに注意されたい.

[3] 手元の計算機のプロセッサは「Intel(R) Core(TM)2 Duo CPU P8600 @ 2.40GHz」である. FORTRAN コンパイラは gfortran (GNU Fortran (Debian 4.4.5-8) 4.4.5) を用いた.

という結果となった. プログラム例は単精度, 倍精度それぞれ ex3_ e.f90, ex3_ d.f90 を参照されたい. なお, 正答は,

$$\sum_{n=0}^{10}(0.1n)^2 = 3.85 \tag{1.12}$$

であり, これは数列の和の公式,

$$\sum_{k=0}^{n}k^2 = \frac{1}{6}n(n+1)(2n+1) \tag{1.13}$$

から容易に確かめられる[4]. 正答と比較すると, パソコン BASIC, あるいは単精度計算において, 最後の $x = 1.0$ に対する項の加算をしないらしいことがうかがえる.

---

[4](1.13) を導出する. 恒等式,

$$(k+1)^3 - k^3 = 3k^2 + 3k + 1$$

を考え, k に $1, 2, \cdots, n$ を代入し, n 個の式の辺辺を加えると,

$$(n+1)^3 - 1 = 3(1^2 + 2^2 + \cdots + n^2) + 3(1 + 2 + \cdots + n) + n$$

となる. よって,

$$\begin{aligned} 3(1^2 + 2^2 + \cdots + n^2) &= (n+1)^3 - 3(1 + 2 + \cdots + n) - (n+1) \\ &= (n+1)^3 - 3\frac{n(n+1)}{2} - (n+1) \\ &= \frac{n(2n+1)(n+1)}{2}. \end{aligned}$$

ゆえに,

$$\sum_{k=0}^{n}k^2 = \frac{1}{6}n(n+1)(2n+1).$$