1.1 例3

$x = 0.0, 0.1, \cdots, 0.9, 1.0$ に対して x^2 を計算して合計する.

結果は下の表のようになりパソコン BASIC では $x = 1.0$ に対する項の加算をしない. この原因は次に説明する浮動小数点による数の表現に依る.

	計算結果
大型機 FORTRAN	3.85
パソコン BASIC	2.85

尚, 正答は

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

であり, これは

$$\sum_{k=0}^{n} k^2 = \frac{1}{6} n(n+1)(2n+1) \tag{1.1.1}$$

を利用すれば容易に確かめられる. この式 (1.1.1) の導出は以下の恒等式を用いることで求められる.

$$(k+1)^3 - k^3 = 3k^2 + 3k^2 + 1 \tag{1.1.2}$$

k に $1, 2, 3, \cdots, n$ を代入し, n 個の式を辺々加えると

$$(n+1)^3 - 1 = 3(1^2 + 2^2 + 3^2 + \cdots + n^2) + 3(1 + 2 + 3 + \cdots + n) + n \tag{1.1.3}$$

よって

$$\begin{aligned} 3(1^2 + 2^2 + 3^2 + \cdots + n^2) &= (n+1)^3 - 3(1 + 2 + 3 + \cdots + n) - (n+1) \\ &= (n+1)^3 - \frac{3n(n+1)}{2} - (n+1) \\ &= \frac{n(2n+1)(n+1)}{2} \end{aligned} \tag{1.1.4}$$

故に

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

1.2　浮動小数点

計算機の中での実数の表現は“ 浮動小数点 ”の形であらわされる.β 進法の場合, 全ての数は f_i を 0 から $\beta-1$ までの整数で $f_1 \neq 0$ とし,E を 0 または正の整数として

$$\pm(0.f_1f_2\cdots f_m)_\beta \times \beta_{10}{}^{\pm E} \tag{1.2.5}$$

という形で表すことができる[1)]. ここで $\pm(0, f_1f_2\cdots f_m)_\beta = \pm\{(f_1)_\beta\beta_{10}{}^{-1}+(f_2)_\beta\beta_{10}{}^{-2}+\cdots+(f_m)_\beta\beta_{10}{}^{-m})\}$ を仮数部 (mantissa),$\pm E$ のことを指数部 (exponent) と呼んでいる.

1.3　表現誤差

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,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. 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, 当然表現には誤差が生じてくる. 以下に 32 ビット語の場合の代表的な例を挙げる.

1.3.1　例 1) IBM 方式

IBM 方式とは次のとおりである.

- $f_1 \neq 0$; 各 f_i は 4 ビットで 0〜F のいずれか
- 指数部 7 ビットを用いて 0〜127 を表せるが, これを $E=-64$〜$+63$ に対応させる
- “ 0 ”はこの表現にはなじまない異質な数なので, 例えば $64+E=0$ を 0 に当てる

式 (1.2.1) において $\beta=16$,$m=6$ とし, 丸め[2)]を切り捨て方式にした数の表現方式を IBM 方式という. この時表現誤差は $f_1=\cdots=f_6=F$ のとき最も小さく

$$\frac{1}{FFFFFF} \sim 16^{-6} \sim 6\times10^{-8}$$

[1)] 添字 β と 10 はそれぞれ β 進法表記と 10 進法表記であることを表す
[2)] 丸めとは切り捨てなどの端数処理のこと.

となる[3]. また,$f_1 = 1$,$f_2 = \cdots = f_6 = 0$ のときもっとも大きく

$$\frac{1}{100000} \sim 16^{-5} \sim 10^{-6}$$

である.

1.3.2　例 2) 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

- $f_1 = 1$ 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 $E = -128 \sim +127$ に対応させる
- 仮数部の符号ビットは“ + ”のとき 0,“ - ”のとき 1
- IBM 方式と同様の理由から, $128 + E = 0$ を 0 に当てる

式 (1.2.1) 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. この時, 表現の相対誤差は $(f_1 =)f_2 = \cdots = f_{24} = 1$ のとき最小で

$$\frac{0.1}{111111111111111111111111} \sim 2^{-25} \sim 3 \times 10^{-8}$$

となる. 最大になるのは $(f_1 = 1)$,$f_2 = \cdots = f_{24} = 0$ のときで

$$\frac{0.1}{100000000000000000000000} \sim 2^{-24} \sim 6 \times 10^{-8}$$

である.

[3]例えば,$FFFFFF.f_\beta$ という数があったとすると, 小数点以下は切り捨てられるので $FFFFFF$ となる. この時生じる誤差は高々1 と見積もれるので, 相対誤差は定義から $\frac{1}{FFFFFF}$ と計算される. 尚,$(FFFFFF)_{16} = (16777215)_{10} = (111111111111111111111111)_2$ である.