桁落ちしない方法：二分木法

桁落ちを起こさない方法としては二分木法も理論的には良いとされている.

二分木法とは端から順に和をとるのではなく, 被加項を二つずつペアにして和をとり, それらをさらに二つずつペアにして和をとるということを繰り返す方法[1)]である. 二分木法は被加項が $x_1, x_2, \cdots$ と順々に読み込まれるときでも, 必要なメモリーは $\log_2 n$ 程度で済む[2)]

桁落ちしない方法：オフセット

二分木法はかなり有効な方法であるしかし, 実際ただの計算で二分木法程の本格的なプログラムを書くのは大変である. よって本当の桁落ちしない方法は, 丸め誤差の観点を重視して算法を設計することである. 1 円玉の例で言うと x_i に $x_i - 1.0000$ を用いて計算すればよい. こうすることによって桁繰り上がりによる桁落ちを起こさないようにしている. これをオフセットをとるという.

[1)] 各項似たもの同士の足し算をすることで差が大きくならないようにして桁落ちを防いでいる.

[2)] n 個の情報があった時

$$\begin{aligned} n &= \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \cdots \quad \frac{n}{2^k} \\ &= \sum_{k=1}^{\frac{n}{2^k} \leq 1} \frac{n}{2^k} \end{aligned}$$

となる. 最後の項の数字は絶対に 1 以下になるので $n \geq 2^k$ の範囲で k の最大値がメモリの量になる.

$$\frac{n}{2^k} \leq 1$$

$$n \leq 2^k$$

$$\log_2 n \leq k$$

よって, 必要なメモリは $\log_2 n$ だが実際に 2 の乗数のときは 1 メモリ消費が増えるので程度としている.

第4章　浮動小数点操作

4.1　実数の数体系

4.1.1　実数の数体系の定義

実数の数体系を

$$x = 0 \quad or \quad s \times b^e \times \sum_{k=1}^{p} f_k \times b^{-k} \tag{4.1.1}$$

と定義する. ここで b および p は 2 以上の整数, f_k は b 未満の非負の整数 (ただし, $f_1 \neq 0$ [1]), s は -1 または, $+1$ であり, e は, ある最小値の整数 e_{min} からある最大値の整数 e_{max} までの間の整数である. これは実数 x を指数部 b^e と小数部 $\sum_{k=1}^{p} f_k \times b^{-k}$ に分けて表現する正規化した浮動小数点表現である[2].

4.1.2　32bit 2 進数での表現

32bit 2 進数の場合

$$x = 0 \quad or \quad s \times 2^e \times \left(\frac{1}{2} + \sum_{k=2}^{24} f_k \times 2^{-k} \right) \tag{4.1.2}$$

---

[1] 常に小数部を小数点以下 1 桁からの表現にするため

[2] 伊理正夫著 数値計算の常識の式 (1) を (4.1.1) の形で表すと

$$\begin{aligned} x &= \pm(0.f_1 f_2 \cdots f_m)_\beta \beta^{\pm E} \\ &= 0 \quad or \quad s \times \beta^{\pm E} \times \sum_{k=1}^{m} (f_k \times \beta^{-k}). \end{aligned}$$

となり, 仮数部が小数部と同じで指数部はそのまま指数部として表している.

となる[3]. このとき $-125 \leq e \leq 128$ [4]である.

4.1.3　組み込み関数での表現

現在使用しているのコンピュータの式 (4.1.1) での e_{max}, e_{min}, b の値および指数を 10 進数に換算した値はそれぞれ数値問合せ組み込み関数 MAXEXPONENT, MINEXPONENT, RADIX および RANGE によって出力できる. (4.1.1) の p の値は数値問合せ組み込み関数 DIGITS と PRECISION によって知ることができる. これらの値から基本実数型の数体系で扱える数の範囲が決まる. また, 最大の数および最小の正の数は数値問合せ組み込み関数 HUGE および TINY によって知ることができる.

4.1.4　マシンイプシロン

p の値が有限であることにより, 1 に足したときに $1 + \epsilon$ が評価されるぎりぎりの数マシンイプシロンが存在する. 数値問合せ組み込み関数 EPSILON によって知ることができる. 関数 epsilon(x) での値は x の数体系でのマシンイプシロン b^{1-p} [5]となる.

---

[3] 項に $\frac{1}{2}$ が入るのは $f_1 \neq 0$ だから 0 か 1 しかない 2 進数の場合 1 に固定だから

[4] 8bit 分なので-127 から 128 まで表現できるはずだが ∞ と notnumber を表すために 2 つ少なくなっている.

[5]
$$1 = (0.1 f_2 \cdots f_p)_b \times b^1$$
この値に桁落ちを起こさないぎりぎりの数 ϵ は
$$\begin{aligned} \epsilon &= (0.00\cdots1)_b \times b^1 \\ &= (0.100\cdots0)_b \times b^{1-(p-1)} \\ &= \frac{1}{b} \times b^{1-(p-1)} \\ &= b^{1-p} \end{aligned}$$
となり, b^{1-p} と同じになる.