8　浮動小数点

浮動小数点とは、指数表現で数を表すことである。

8.1　符号, 指数, 仮数

科学技術計算では膨大な桁数の数字が必要となる。例えば、

光の速さ: 2.99792458×10^8 m/sec

アボガドロ定数: 6.022×10^{23} [mol^{-1}、個]

電気素量: 1.602×10^{-19} [C]

プランク定数: 6.626×10^{-34} [J・s]

である。これらを固定小数点数で表現しようとすれば、整数の桁に 40 個，小数の桁に 40 個，合わせて 80 桁のデータサイズが必要になる。しかし、この 80 桁全てに数字が必要なわけではない。有効数字は 8-10 桁程度である。このことから，詳細な数値は誤差の範囲で無意味、即ち 80 桁中に意味のある数は 8 から 10 個以下程度であり，残りの 70 桁はすべて 0 となる。これでは不経済である。そこで，もう少しコンパクトな表現が考案された。これを浮動小数点表示という。

浮動小数点数では，有効数字 (significand) を 8 桁程度にして，残りを指数 (exponent) で表現する。有効数字を表す有意な数は，(表現の一意性を保つために) 整数部が一桁になるように正規化する。これを仮数 (mantissa) と呼ぶ。

浮動小数点は、「符号、指数、仮数」のビット列で表現される。

8.2　IEEE 754

浮動小数点数の表現の一つである IEEE（Institute of Electrical and Electronics Engineers; 米国電子技術者協会、アイトリプルイーと発音する）754 規格では，浮動小数点数を 32 ビットまたは 64 ビットで表現するように定められている。32 ビットのことを単精度 (Single Precision) と呼び，64 ビットのことを倍精度 (Double Precision) と呼ぶ。

IEEE 754 では、ビットの使い方として，符号 (sign)，指数 (exponent)，仮数 (mantissa) を以下のように定義している。

- 有為な数を仮数と呼び、残りのビットは指数部、符号ビットとする。
- 32 ビット表現を単精度と呼び，符号ビットを 1 ビット，仮数部を 23 ビット，指数部を 8 ビットとする。
- 64 ビット表現を倍精度と呼び，符号ビットを 1 ビット，仮数部を 52 ビット, 指数部を 11 ビットとする。

例えば、光の速さ 2.99792458×10^8 [m/sec.] では，2.99792458 が仮数であり，8 が指数である。

8.3 IEEE 754 のビット表現

単精度の場合のビット表現を解説する。

単精度の IEEE 754 浮動小数点数は，符号ビット (1 ビット)，指数部 (8 ビット)，仮数部 (23 ビット) の計 32 ビットで表現される。

指数部は次の表のようになる。

1000	0011	+4 乗
1000	0010	+3 乗
1000	0001	+2 乗
1000	0000	+1 乗
0111	1111	0 乗
0111	1110	-1 乗
0111	1101	-2 乗
0111	1100	-3 乗
0111	1011	-4 乗

指数部は 2 の 0 乗のとき，0111 1111 となる。

仮数部は 2 を基数として整数部が 1 桁になるように正規化した数の 2 進数表現である。

正規化によって仮数部の最上位ビットは常に 1 になる。そのため、実際に用意しておく必要はなくなる。

倍精度の 52 ビットであれば、最上位の 1 を隠れビット hidden bit として表現に含めなければ，53 ビット分の情報が表現できることになる。

表 10: 小数の二進数表現

0.1000	$1/2$	$= 0.5$
0.0100	$1/(2 \times 2)$	$= 0.25$
0.0010	$1/(2 \times 2 \times 2)$	$= 0.125$
0.0001	$1/(2 \times 2 \times 2 \times 2)$	$= 0.0625$

符号ビットは 1 ビットであり，1 は負の数，0 は正の数を表す。

8.4 IEEE 754 フォーマットの例

例えば，$21.6875_{(10)} = (00010101.1011)_{(2)} = (1.01011011)_{(2)} \times 2^4$ となるので、指数部の 4 乗が 10000011 となり，仮数部は 1.01011011000000000000000 となる。仮数部の最上位ビットの 1 を隠れビットにして含めなければ，結果として，単精度では

”0 —1000 0011 — 010 1101 1000 0000 0000 0000” となる。

次のプログラム (ファイル名は `FloatingPoint.java` とする) で，10 進数の IEEE 754 浮動小数点表現が得られる。

FloatingPoint.java

```
class FloatingPoint {
    public static void main(String[] args) {
        float f = Float.parseFloat(args[0]);
        String str = Integer.toBinaryString(Float.floatToRawIntBits(f));
        System.out.println(args[0] + " = " + str);
    }
}
```

```
$ javac FloatingPoint.java

$ java FloatingPoint 21.6875
21.6875 = 10000011010110110000000000000000

$ java FloatingPoint -21.6875
-21.6875 = 11000001101011011000000000000000

$ java FloatingPoint 1
1 = 11111110000000000000000000000000

$ java FloatingPoint 0.5
0.5 = 11111100000000000000000000000000
```