Chapter 5

Floating point arithmetic.

- Binary representation and base 2 arithmetic.

Addition.

$$\begin{array}{r} 1010 \\ +\ 11011 \\ \hline 100101 \end{array} \qquad \begin{array}{r} 10 \\ +\ 27 \\ \hline 37 \end{array}$$

Subtraction.

Multiplication:

Shift and add.

Division.

~~Dec~~ Rational Numbers.

$0.b_1 b_2 b_3 \ldots$

$= b_1 \times 2^{-1} + b_2 \times 2^{-2} + \ldots$

What is rational in base 10 may not be rational in base 2.

e.g. 0.1 in base 10 is $0.0\overline{001100}$

Floating point representation.

$$\pm m \times 2^E \qquad 1 \le m < 2$$

$$10 = 1010_2 = 1.010_2 \times 2^3$$

Three fields.

Sign

mantissa

exponent.

Float: 32 bits.

1 bit for sign. $\begin{cases} 0 : \text{positive} \\ 1 : \text{negative.} \end{cases}$

8 bits for exponent

23 bits for mantissa or significand.

eg. $10 = 1.010_2 \times 2^3$

0	E = 3	m: 1.010 0

$5.5 = 1.011 \times 2^2$

0	E = 2	1.0110 . . . 0

Some optimizations.

$m \times 2^{E}$

$1 \leq m < 2$

we always know that the most significant bit is 1. So there is no need to store it.

$\therefore$ $10 = 1.010 \times 2^{3}$ is stored as

0	E = 3	010 0

This is called hidden bit representation.

Rounding and approximation.

Consider the number

$$\frac{1}{10} = 1.\overbrace{10011001100110011001100}^{23 \text{ bits}} \times 2^{-4}$$

This is an approximation.

It can be written as

0	E = -4	10011001100 . . . 1100

or

1	E = -4	1011001100 . . . 1101

Question: How to represent 0?

Machine precision.

Gap between 1 and next number is called machine precision.

The next number is

0	E=0	000 - - - - - - - - - 1

$$= 1 + 2^{-23}$$

Machine precision $= 2^{-23}$ or 1.2×10^{-7}

For double precision.

1 bit for sign.

11 bits for exponent.

52 for mantissa.

$\therefore$ double precision machine precision is

$$2^{-52} \simeq 2.2\times10^{-16}$$

Gap between 0 and smallest non-zero number and next number.

0 , 1.0×2^{-E}

next smallest is $(1+\epsilon)\times 2^{-E}$

Gap is $\epsilon \times 2^{-E}$ (ϵ is machine precision)

IEEE Floating Point Arithmetic.

Representation for 0, $\pm\infty$ and NaN

- Special bits in the exponent field.
- Also used to represent subnormal numbers.

Three standard precision.

32 bit : 1 sign, 8 exponent, 23 significand.

64 bit : 1 sign, 11 exponent, 52 significand.

80 bit : 1 sign, 15 exponent, 64 significand.

Exponent.

00 --- 0 $\quad \pm .b_1 b_2 \ldots b_{52} \times 2^{-1022}$ 0 or Subnormal.

00 .... 1 $\quad \pm 1.b_1 b_2 \ldots b_{52} \times 2^{-1022}$

00 ... 10 $\quad \pm 1.b_1 b_2 \ldots b_{52} \times 2^{-1021}$

Exponent field is actual exponent + 1023

1111 ... 1 $\quad \pm\infty$ if $b_1 b_2 \ldots b_{52} = 0$

NAN otherwise.

Smallest Subnormal number.

$$\underbrace{0}_{s} \quad \underbrace{00\cdots 0}_{E} \quad \underbrace{00\cdots 1}_{m}$$

$$2^{-1022+0} \times 2^{52} = 2^{-1074}$$

$$0: \quad 0 \quad \underbrace{00\cdots 0}_{E} \quad \underbrace{00\cdots 0}_{m}.$$

Rounding:

- Round down
- Round up.
- Round towards 0
- Round to nearest.

Default is round to nearest.

$\frac{1}{10} = 1.100\overline{1100} \times 2^{-4}$ is replaced by

0 01111111011 10110011001 1 1001

(round to nearest)

0 01111111011 10110011001 1 1010

(round down or round towards 0).

Absolute rounding error

$$|\text{round}(x) - x|$$

in double precision, if

$$x = \pm(1.b_1 b_2 \ldots b_{53} \ldots) \times 2^E$$

E is in the range -1022 to 1023.

then absolute rounding error $< 2^{-52} \times 2^E$

for any rounding mode.

Relative rounding error

$$\frac{|\text{round}(x) - x|}{|x|} \leq \epsilon \quad \cancel{(\ldots)}.$$

(Here ϵ is machine precision)

$\therefore$ we can write

$$\text{round}(x) = x(1+\delta) \qquad |\delta| < \epsilon$$

(or $\frac{\epsilon}{2}$ for round to nearest).

$$a \oplus b = \text{round}(a+b) = (a+b)(1+\delta_1)$$

$$a \ominus b = \text{round}(a-b) = (a-b)(1+\delta_2)$$

$$a \otimes b = \text{round}(a \times b) = (a \times b)(1+\delta_3)$$

$$a \oslash b = \text{round}(a/b) = (a/b)(1+\delta_4)$$

$$|\delta_i| < \epsilon$$