Floating Point Representation

[Computational Physics]

Floating Decimal Point : Scientific Form

$$256.78 \text{ is written as} +2.5678\times10^{2}$$

$$0.003678 \text{ is written as} +3.678\times10^{-3}$$

$$-256.78 \text{ is written as} -2.5678\times10^{2}$$

Example

The form is

$$\text{sign} \times \text{mantissa} \times 10^{\text{exponent}}$$

or

$$\sigma \times m \times 10^{e}$$

Example: For

$$-2.5678 \times 10^{2}$$

$$\sigma = -1$$

$$m = 2.5678$$

$$e = 2$$

Floating Point Format for Binary Numbers

$$y = \sigma \times m \times 2^{e}$$

$$\sigma = \text{sign of number} \quad (0 \text{ for } +\text{ve}, 1 \text{ for } -\text{ve})$$

$$m = \text{mantissa} \quad \left[(1)_2 < m < (10)_2\right]$$

1 is not stored as it is always given to be 1.

$$e = \text{integer exponent}$$

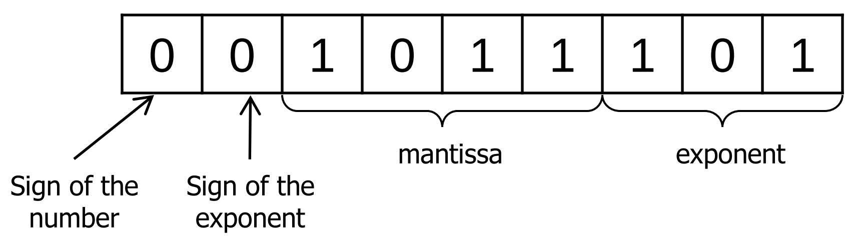
Example

9 bit-hypothetical word

- the first bit is used for the sign of the number,
- the second bit for the sign of the exponent,
- the next four bits for the mantissa, and
- the next three bits for the exponent

$$(54.75)_{10} = (110110.11)_2 = (1.1011011)_2 \times 2^5$$
$$\cong (1.1011)_2 \times (101)_2$$

We have the representation as



Machine Epsilon

Defined as the measure of accuracy and found by difference between 1 and the next number that can be represented

Example

Ten bit word

- Sign of number
- Sign of exponent
- Next four bits for exponent
- Next four bits for mantissa

0	0	0	0	0	0	0	0	0	0

$= (1)_{10}$

Next number →

0	0	0	0	0	0	0	0	0	1

$= (1.0001)_2 = (1.0625)_{10}$

$$\in_{mach} = 1.0625 - 1 = 2^{-4}$$

Relative Error and Machine Epsilon

The absolute relative true error in representing a number will be less then the machine epsilon

Example

$$(0.02832)_{10} \cong (1.1100)_2 \times 2^{-5}$$

$$= (1.1100)_2 \times 2^{-(0110)_2}$$

10 bit word (sign, sign of exponent, 4 for exponent, 4 for mantissa)

0	1	0	1	1	0	1	1	0	0
Sign of the number	Sign of the exponent	exponent				mantissa			

$$(1.1100)_2 \times 2^{-(0110)_2} = 0.0274375$$

$$\in_a = \left| \frac{0.02832 - 0.0274375}{0.02832} \right|$$

$$= 0.034472 < 2^{-4} = 0.0625$$

IEEE 754 Standards for Single Precision Representation

Computational Physics

IEEE-754 Floating Point Standard

- Standardizes representation of floating point numbers on different computers in single and double precision.

- Standardizes representation of floating point operations on different computers.

One Great Reference

What every computer scientist (and even if you are not) should know about floating point arithmetic!

http://www.validlab.com/goldberg/paper.pdf

IEEE-754 Format Single Precision

32 bits for single precision

$$\underbrace{0}_{\text{Sign (s)}}\,\underbrace{0\,0\,0\,0\,0\,0\,0\,0}_{\text{Biased Exponent (e')}}\,\underbrace{0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0}_{\text{Mantissa (m)}}$$

$$\text{Value} = (-1)^s \times (1 \bullet m)_2 \times 2^{e'-127}$$

Example#1

1	1	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sign (s) | Biased Exponent (e′) | Mantissa (m)

$$\text{Value} = (-1)^s \times (1.m)_2 \times 2^{e'-127}$$

$$= (-1)^1 \times (1.10100000)_2 \times 2^{(10100010)_2 - 127}$$

$$= (-1) \times (1.625) \times 2^{162-127}$$

$$= (-1) \times (1.625) \times 2^{35} = -5.5834 \times 10^{10}$$

Example#2

Represent -5.5834×10^{10} as a single precision floating point number.

?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Sign (s) — Biased Exponent (e′) — Mantissa (m)

$$-5.5834 \times 10^{10} = (-1)^{1} \times (1.?) \times 2^{\pm ?}$$

Exponent for 32 Bit IEEE-754

8 bits would represent

$$0 \leq e' \leq 255$$

Bias is 127; so subtract 127 from representation

$$-127 \leq e \leq 128$$

Exponent for Special Cases

Actual range of e'

$$1 \le e' \le 254$$

$e' = 0$ and $e' = 255$ are reserved for special numbers

Actual range of e

$$-126 \le e \le 127$$

Special Exponents and Numbers

$$e' = 0$$ — all zeros

$$e' = 255$$ — all ones

s	e'	m	Represents
0	all zeros	all zeros	0
1	all zeros	all zeros	-0
0	all ones	all zeros	∞
1	all ones	all zeros	$-\infty$
0 or 1	all ones	non-zero	NaN

IEEE-754 Format

The largest number by magnitude

$$(1.1........1)_2 \times 2^{127} = 3.40 \times 10^{38}$$

The smallest number by magnitude

$$(1.00......0)_2 \times 2^{-126} = 2.18 \times 10^{-38}$$

Machine epsilon

$$\varepsilon_{mach} = 2^{-23} = 1.19 \times 10^{-7}$$

Additional Resources

For all resources on this topic such as digital audiovisual lectures, primers, textbook chapters, multiple-choice tests, worksheets in MATLAB, MATHEMATICA, MathCad and MAPLE, blogs, related physical problems, please visit

Computational Physics
/topics/floatingpoint_representation.html

THE END