What Is Floating-Point Arithmetic?

Nicholas J. Higham*

May 4, 2020

A floating-point number system F is a finite subset of the real line comprising numbers of the form

$$y = \pm m \times \beta^{e-t},$$

where β is the base, t is the precision, and $e \in [e_{\min}, e_{\max}]$ is the exponent. The system is completely defined by the four integers β, t, $e_{\min}$, and $e_{\max}$. The *significand* m satisfies $0 \le m \le \beta^t - 1$. *Normalized numbers* are those for which $m \ge \beta^{t-1}$, and they have a unique representation. *Subnormal numbers* are those with $0 < m < \beta^{t-1}$ and $e = e_{\min}$.

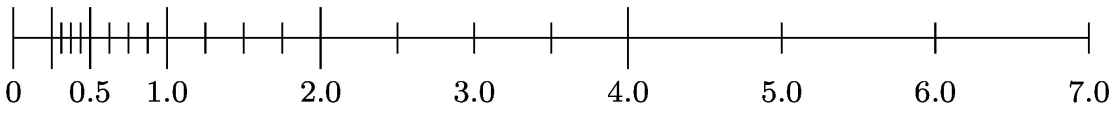
An alternative representation of $y \in F$ is

$$y = \pm\beta^e \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \cdots + \frac{d_t}{\beta^t} \right) = \pm\beta^e \times .d_1 d_2 \ldots d_t,$$

where each digit d_i satisfies $0 \le d_i \le \beta - 1$ and $d_1 \ne 0$ for normalized numbers.

The floating-point numbers are not equally spaced, but they have roughly constant relative spacing (varying by up to a factor β).

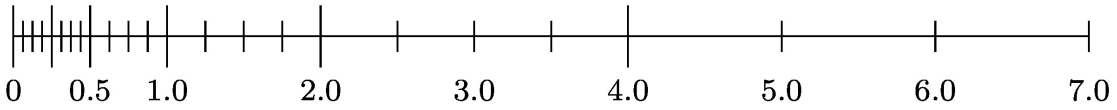
Here are the normalized nonnegative numbers in a toy system with $\beta = 2$, $t = 3$, and $e \in [-1, 3]$.



Three key properties that hold in general for binary arithmetic are visible in this example.

- The spacing of the numbers increases by a factor 2 at every power of 2.
- The spacing of the numbers between 1/2 and 1 is $u = 2^{-t}$, which is called the *unit roundoff*. The spacing of the numbers between 1 and 2 is $\epsilon = 2^{1-t}$, which is called the *machine epsilon*. Note that $\epsilon = 2u$.
- There is a gap between 0 and the smallest normalized number, which is $2^{e_{\min}-1}$. The subnormal numbers fill this gap with numbers having the same spacing as those between $2^{e_{\min}-1}$ and $2^{e_{\min}}$, namely $2^{e_{\min}-t}$. The next diagram shows the complete set of nonnegative normalized and subnormal numbers in the toy system.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).



In MATLAB, `eps` is the machine epsilon, `eps(x)` is the distance from `x` to the next larger (in magnitude) floating-point number, `realmax` is the largest finite number, and `realmin` is the smallest normalized positive number.

A real number x is mapped into F by rounding, and the result is denoted by $\mathrm{fl}(x)$. If x exceeds the largest number in F then we say that $\mathrm{fl}(x)$ overflows, and in IEEE arithmetic it is represented by Inf. If $x \in F$ then $\mathrm{fl}(x) = x$; otherwise, x lies between two floating-point numbers and we need a rule for deciding which one to round x to. The usual rule, known as round to nearest, is to round to whichever number is nearer. If x is midway between two floating-point numbers then we need a tie-breaking rule, which is usually to round to the number with an even last digit. If $x \neq 0$ and $\mathrm{fl}(x) = 0$ then $\mathrm{fl}(x)$ is said to underflow.

For round to nearest it can be shown that

$$\mathrm{fl}(x) = x(1+\delta), \quad |\delta| \leq u.$$

This result shows that rounding introduces a relative error no larger than u.

Elementary floating-point operations, $+$, $-$, $*$, $/$, and $\sqrt{}$ are usually defined to return the correctly rounded exact result, so they satisfy

$$\mathrm{fl}(x \mathrm{\ op\ } y) = (x \mathrm{\ op\ } y)(1+\delta), \quad |\delta| \leq u, \quad \mathrm{op} \in \{+, -, *, /, \sqrt{}\}.$$

Most floating-point arithmetics adhere to the IEEE standard, which defines several floating-point formats and four different rounding modes.

Another form of finite precision arithmetic is fixed-point arithmetic, in which numbers have the same form as F but with a fixed exponent e, so all the numbers are equally spaced. In most scientific computations scale factors must be introduced in order to be able to represent the range of numbers occurring. Fixed-point arithmetic is mainly used on special purpose devices such as FPGAs and in embedded systems.

References

This is a minimal set of references, which contain further useful references within.

- D. Goldberg, What Every Computer Scientist Should Know About Floating-Point Arithmetic, ACM Computing Surveys 23, 5–48, 1991.
- Nicholas J. Higham, Accuracy and Stability of Numerical Algorithms, second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019 (Revision of IEEE 754-2008), The Institute of Electrical and Electronics Engineers, New York, 2019.
- Jean-Michel Muller, Nicolas Brunie, Florent de Dinechin, Claude-Pierre Jeannerod, Mioara Joldes, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, and Serge Torres, Handbook of Floating-Point Arithmetic, second edition, Birkhäuser, Boston, MA, 2018.

Related Blog Posts

- A Multiprecision World (2017)
- Book Review Revisited: Overton's Numerical Computing with IEEE Floating Point Arithmetic (2014)
- Half Precision Arithmetic: fp16 Versus bfloat16 (2018)
- The Rise of Mixed Precision Arithmetic (2015)
- What Is IEEE Standard Arithmetic?—forthcoming
- What Is Rounding? (2020)

This article is part of the "What Is" series, available from `https://nhigham.com/category/what-is` and in PDF form from the GitHub repository `https://github.com/higham/what-is`.