Chapter 2

Some Basic Things About Computer Arithmetic

2.1 Floating-Point Arithmetic

The aim of this section is to provide the reader with some basic concepts of floating-point arithmetic, and to define notations that are used throughout the book. For further information, the reader is referred to Goldberg’s paper [88], which gives a good survey of the topic, and Kahan’s lecture notes [106], which offer interesting and useful information. Further information can be found in [19, 39, 35, 40, 96, 111, 116, 148, 193, 197]. Here we mainly focus on the IEEE-754 standard [4] for radix-2 floating-point arithmetic. The IEEE standard (and its follower, the IEEE-854 radix-independent standard) was a key factor in improving the quality of the computational environment available to programmers. Before the standard, floating-point arithmetic was a mere set of cooking recipes that sometimes worked well and sometimes did not work at all.¹

2.1.1 Floating-point formats

In a floating-point system of radix (or base) $r$, mantissa length $n$, and exponent range $E_{\text{min}} \ldots E_{\text{max}}$, a number $x$ is represented by a mantissa, or significant $M_x = x_0.x_1x_2 \cdots x_{n-1}$ which is an $n$ radix $r$-digit number satisfying $0 \leq M_x < r$, a sign $s_x = \pm 1$, and an exponent $E_x$, $E_{\text{min}} \leq E_x \leq E_{\text{max}}$, such that

$$x = s_x \times M_x \times r^{E_x}.$$

¹We should mention a few exceptions, such as some HP pocket calculators and the Intel 8087 co-processor, that were precursors of the standard.
For accuracy reasons it is frequently required that the floating-point representations be normalized, that is, that the mantissas be greater than or equal to 1. This requires a special representation for the number zero.

An interesting consequence of that normalization, for radix 2, is that the first mantissa digit of a floating-point nonzero number must always be "1." Therefore there is no need to store it, and in many computer systems, it is actually not stored (this is called the "hidden bit" or "implicit bit" convention). Table 2.1 gives the basic parameters of the floating-point systems that have been implemented in various machines. Those figures have been taken from references [96, 106, 111, 148]. For instance, the largest representable finite number in the IEEE-754 double precision format [4] is

\[(2 - 2^{-52}) \times 2^{1023} \approx 1.7976931348623157 \times 10^{308}\]

and the smallest positive normalized number is

\[2^{-1022} \approx 2.225073858507201 \times 10^{-308}.\]

Arithmetic based on radix 10 is used in pocket calculators, whereas almost all other current computing systems use base 2. Various studies [19, 35, 116] have shown that radix 2 with the hidden bit convention gives better accuracy than all other radices (this does not imply that operations — e.g., divisions — cannot benefit from being done in a higher radix inside the arithmetic operators).

### 2.1.2 Rounding modes

Let us define a machine number to be a number that can be exactly represented in the floating-point system under consideration. In general, the sum, the product, and the quotient of two machine numbers is not a machine number and the result of such an arithmetic operation must be rounded.

In a floating-point system that follows the IEEE standard, the user can choose an active rounding mode from:

- rounding towards \(-\infty\): \(\nabla (x)\) is the largest machine number less than or equal to \(x\);
- rounding towards \(+\infty\): \(\Delta (x)\) is the smallest machine number greater than or equal to \(x\);

---

2 A major difference between computers and pocket calculators is that usually computers do much computation between input and output of data, so that the time needed to perform a radix conversion is negligible compared to the whole processing time. If pocket calculators used radix 2, they would perform radix conversions before and after almost every arithmetic operation. Another reason for using radix 10 in pocket calculators is the fact that many simple decimal numbers such as 0.1 are not exactly representable in radix 2.