1.8 Representation of numbers: Truncation and rounding

On a computer, the normalized binary form of numbers is used. The precision of such a number is limited due to the final amount of bits available to store it. Therefore, most floating-point numbers used by a computer are only approximations of the true values.

How does the finite amount of available bits translate to a precision expressed in decimal places? How to minimize the deviation between the true value and the approximation?

While the exponent only determines the range of accessible numbers, the precision is related to the smallest difference in neighboring numbers that can be resolved by the mantissa, $(\Delta\bar{x})_{\min}$. This obviously refers to the last place of the mantissa. Smaller differences that correspond to places beyond that cannot be stored.

The bits available for the whole number are used as follows: one bit for the sign, usually 8 or 11 bits for the exponent, and the remaining bits for the fractional part of the mantissa (since the leading digit is always 1 for normalized binary numbers, it does not have to be stored but is assumed implicitly). If there are m bits to store the mantissa, then its last place corresponds to the following value [cf. Eq. (1.5)]:

$$(\Delta\bar{x})_{\min} = \underbrace{0.00...01}_{m+1 \text{ places}} = 2^{-m}. \tag{1.11}$$

As an example, consider a data type using 32 bits per number (usually called *single precision*) as follows: one bit for the sign, 8 bits for the exponent, and $m = 32 - 9 = 23$ bits for the mantissa. Therefore, $(\Delta\bar{x})_{\min} = 2^{-m} = 2^{-23} \approx 1.2 \times 10^{-7}$, i.e., this data type has a precision of seven decimal places.

In general, the *machine precision* is the distance from 1.0 to the next-largest *double-precision* number, that is `eps` $= 2^{-52} \approx 2.2 \times 10^{-16}$. (Double-precision data consist of 64 bits: one for the sign, 11 for the exponent, and 52 for the mantissa.)

Truncation means that after conversion to the normalized binary form, all places of a number beyond the last place that the mantissa can contain are thrown away. This leads to a largest possible error of $(\Delta\bar{x})_{\min}$ absolute. Rounding to the last place of the mantissa reduces this error by a factor 1/2, i.e. the accuracy increases by one binary place.