1.2 Round-off errors and computer arithmetic

1. Why and how computer arithmetic is different from arithmetic in mathematics.

Computer representation of rational and irrational numbers. Finite-digit arithmetic.

The error produced when a calculator or computer is used to perform real-number calculations is called round-off error.

Binary machine numbers

According to the IEEE standard, 64-bit (binary digit representation) used for a real number consists of a sign indicator (the first bit, denoted s), followed by an 11-bit exponent, c, called the **characteristic**, and by a 52-bit binary fraction, f, called **mantissa**. The base for the exponent is 2. 52 binary digits accommodate between 16 and 17 decimal digits, so we can expect at least 16 decimal digits of precision. The exponent of 11 binary digits gives a range of 0 to $2^{11} - 1 = 2047$. However, actual range of the exponent is from -1023 to 1024, in order to be able represent small numbers.

A floating –point number has the form

$$(-1)^s 2^{c-1023}(1 + f)$$

Ex. 0 10000000011 101110010001.... (all …. are 0s, 40 of them). Then $s = 0$ indicating that the number is positive;

$$c = 1 * 2^{10} + 0 * 2^9 + \cdots + 1 * 2^1 + 1 * 2^0 = 1024 + 2 + 1 = 1027.$$

The exponential part of the number is $2^{1027-1023} = 2^4$. The final 52 bits specify the mantissa

$$f = 1 * \left(\frac{1}{2}\right)^1 + 0 * \left(\frac{1}{2}\right)^2 + 1 * \left(\frac{1}{2}\right)^3 1 * \left(\frac{1}{2}\right)^4 + 1 * \left(\frac{1}{2}\right)^5 + 1 * \left(\frac{1}{2}\right)^8 + 1 * \left(\frac{1}{2}\right)^{12}$$

Thus, the decimal number represented here is

$$(-1)^0 * 2^4 \left(1 + 1 * \left(\frac{1}{2}\right)^1 + 0 * \left(\frac{1}{2}\right)^2 + 1 * \left(\frac{1}{2}\right)^3 1 * \left(\frac{1}{2}\right)^4 + 1 * \left(\frac{1}{2}\right)^5 + 1 * \left(\frac{1}{2}\right)^8 + 1 * \left(\frac{1}{2}\right)^{12}\right) = 27.56640625 .$$

The next smallest machine number is 0 10000000011 101110010000.... (all …. are 1s, 40 of them) and the next largest machine number is

0 10000000011 101110010001....1 (all …. are 0s, 39 of them). In fact, our machine number represents any real number belonging to an interval shown in the middle of p.16 in the text.

The smallest normalized positive number that can be represented has $s = 0, c = 1, and\ f = 0$, meaning

$2^{-1022}(1 + 0) \approx 0.2251 * 10^{-307}$,

And the largest has $s = 0, c = 2046, and\ f = 1 - 2^{-52}$, and is equivalent to

$$2^{1023}(2 - 2^{-52}) \approx 0.17977 * 10^{309}$$

Underflow, overflow.

Decimal machine numbers

To illustrate the computational challenges occurring when using machine numbers, assume that normalized decimal floating-point form is used:

$$\pm 0.d_1 d_2 \ldots d_k \times 10^n, \qquad 1 \le d_1 \le 9, and\ 0 \le d_i \le 9\ for\ 2 \le i \le k.$$

Assume $= 0.d_1d_2\ldots d_kd_{k+1}d_{k+2}\cdots \times 10^n$. The floating-point form of y is obtained by terminating the mantissa of y at k decimal digits. This can be obtained by one of two common ways, **chopping** or **rounding**.

Chopping: $fl(y) = 0.d_1d_2\ldots d_k \times 10^n$.

Rounding: add $5 * 10^{n-(k+1)}$, then chop, which is equivalent to

$$fl(y) = 0.\delta_1\delta_2\ldots\delta_k \times 10^n$$

When $d_{k+1} < 5$, $\delta_i = d_i$ for all $i = 1, \cdots, k$. This is called rounding down. However, when we round up (that is, when $d_{k+1} \geq 5)$, the digits (and even the exponent) might change.

Ex. 3 from the text.

$x = \frac{5}{7}, y = \frac{1}{3}$.

Five-digit arithmetic with chopping

$$x = \frac{5}{7} = 0.\overline{714285};\ y = \frac{1}{3} = 0.\bar{3}$$

$$fl(x) = 0.71428 * 10^0; fl(y) = 0.33333 * 10^0$$

$$x \oplus y = fl\big(fl(x) + fl(y)\big) = fl(0.71428 * 10^0 + 0.33333 * 10^0)$$
$$= fl(1.04761 * 10^0) = 0.10476 * 10^1$$

$$x + y = \frac{5}{7} + \frac{1}{3} = \frac{22}{21}$$

Abs. error = $\left|\frac{22}{21} - 0.10476 * 10^1\right| = 0.19 * 10^{-4}$

Relative error = $\frac{0.19*10^{-4}}{22/21} = 0.182 * 10^{-4}$.

Definition (1.15)

Suppose that p^* is an approximation to p. The **actual error** is $p - p^*$, the **absolute error** is $|p - p^*|$, and the **relative error** is $\dfrac{|p - p^*|}{|p|}$, provided that $p \neq 0$.

Definition (1.16)

The number p^* is said to approximate p to t **significant digits** (or figures) if t is the largest nonnegative integer for which

$$\frac{|p - p^*|}{|p|} \leq 5 \times 10^{-t}.$$

Example 5. Let $p = 0.54617\ and\ q = 0.54601$. Use four-digit arithmetic to approximate $p - q$ and determine absolute and relative errors using (a) rounding and (b) chopping.

Exact $r = p - q = 0.00016$.

With rounding, $p^* = 0.5462;\ q^* = 0.5460;\ r^* = 0.0002\ (small)$;

$$rel.error = \frac{|0.00016 - 0.0002|}{|0.00016|} = 0.25\ (large).$$

With chopping, $p^* = 0.5461;\ q^* = 0.5460;\ r^* = 0.0001\ (small)$;

$$rel.error = \frac{|0.00016 - 0.0001|}{|0.00016|} = 0.375\ (large).$$

Is it possible to avoid loss of accuracy due to round-off error when performing finite-digit arithmetic?

Example: Solving quadratic equations by quadratic formula.

Solve $x^2 + 62.10x + 1 = 0$. Use four-digit arithmetic with rounding.

The solutions are approximately $x_1 = -0.01610723$ and $x_2 = -62.08390$

(1)Compute $\sqrt{b^2 - 4ac}$ = $\sqrt{(62.10)^2 - (4.000)(1.000)(1.000)}$ = $\sqrt{3856. - 4.000}$ = $\sqrt{3852}$ = 62.06.

$fl(x_1) = \frac{-62.10+62.06}{2.000} = -0.02000$;

Rel. error = $\frac{|-0.01611+0.02000|}{|-0.01611|} \approx 2.4 * 10^{-1}$ (large).

Rel. error in x_2 is approximately $3.2 * 10^{-4}$.

How to obtain a better four-digit rounding approximation for x_1 ? Use an alternative variant of quadratic formula:

$x_1 = \frac{-2c}{b+\sqrt{b^2-4ac}}$. Then $fl(x_1) = -0.01610$, with a small rel. error of $6.2 * 10^{-4}$. However, the rel. error in x_2 now becomes $2.4 * 10^{-1}$ (large).

Be aware of a potential source of large relative errors!