Basic concepts of numerical methods

Number representations

The smallest addressable unit is usually an 8 bit byte (except in some word based machines, like Cray).

Integer, usually 2 bytes, one bit reserved for the sign. Then the absolute value of an integer is at most $2^{15} - 1 = 32\ 767$.

Often also double integers, 4 bytes. Largest value $2^{31} - 1 = 2\ 147\ 483\ 647$.

Real number usually 4 bytes.

- sign
- coefficient (mantissa), normalized in the range 0.1 – 1
- exponent: sign + absolute value or 2's complement

E.g. on a PC the mantissa of a single precision real number consists of 23 bits. The bit following the decimal point is always 1, and need not be stored. Precision 24 bits or $\log_{10} 2^{24} \approx 7$ decimals.

Exponent part 8 bits, $e = 127$ (bias) + true exponent; $1 \leq e \leq 254$. Range of values $2^{-126} \approx 10^{-38}$ – $2^{128} \approx 3 \times 10^{38}$.

Double precision: about twice as many significant digits and wider range of values. On a PC the precision is about 15 decimals, and range about $10^{-308} \ldots 10^{308}$. (If this range is not sufficient, you should probably rethink your algorithm ...)

Finite precision

Precision and floating point (real number) representations are described by machine constants.

"Machine epsilon" ϵ is the smallest number that added to one produces a sum bigger than one:

$$\epsilon = \min\{x|1 + x > 1\}.$$

On a PC the machine constant of double precision real numbers is $\epsilon = 2.2 \times 10^{-16}$.

NB! this is very much bigger than the smallest representable positive value.

PC's and some other systems use the IEEE extended arithmetic that includes two special values, Inf (infinity) and NaN (Not a Number).

Division by zero gives Inf, which can be, with caution, used in further calculations. However, it indicates that the algorithm is not quite decent, and in some systems the program may crash. You should find the source of the potential problem.

Operations like 0/0, 0×Inf and Inf−Inf are indeterminate, and the value is NaN. All further operations with such a value will also give NaN. That certainly means that your algorithm is not working properly.

Errors

If the exact value is a, the absolute error of its approximate value $\tilde{a}$ is

$$\Delta a = \tilde{a} - a.$$

The relative error is

$$e = \frac{\Delta a}{a} = \frac{\tilde{a} - a}{a}.$$

Often a is not known, but Δa can be estimates by e.g. its statistical properties. An estimate of the relative error is then

$$e \approx \frac{\Delta a}{\tilde{a}}.$$

Rounding:

1.490	1, 1.5
1.551	2, 1.6
1.500	2
2.500	2

Truncation: the least significant part of the number is discarded.

This is the way real numbers are converted to integers in Fortran.

Error propagation in arithmetic operations

Addition

In addition also the errors are added. If the signs of the errors are random, the errors partly cancel each others.

$$1.57 + 0.76 = 2.33.$$

Calculate the sum by rounding the numbers to one decimal place:

$$1.6 + 0.8 = 2.4$$

The relative errors of the terms are

$$\frac{1.6 - 1.57}{1.57} = 0.019, \quad \frac{0.8 - 0.76}{0.76} = 0.053.$$

The relative error of the sum is

$$\frac{2.4 - 2.33}{2.33} = 0.030,$$

The relative error of the sum can never exceed the largest relative error of the positive terms.

Change the example a little:

$$1.57 + 0.74 = 2.31.$$

$$1.6 + 0.7 = 2.3$$

Now the relative errots of the terms are

$$\frac{1.6 - 1.57}{1.57} = 0.019, \quad \frac{0.7 - 0.74}{0.74} = -0.054.$$

The relative error of the sum is

$$\frac{2.3 - 2.31}{2.31} = -0.004.$$

Addition is a smoothing operation, if the signs of the errors of individual terms are random.

Addition can cause problems if the magnitudes of the terms are very different. If the precision is 7 decimals, $1.0 + 3 \times 10^{-8} = 1.0$.

When evaluating a long series it may be useful to calculate first the sum of the smallest terms, or clump the terms into groups in such a way that the sum in each group is of the same magnitude.

Subtraction

In mathematics, subtraction is not essentially different from addition. When calculating with finite accuracy the difference is crucial.

Two approximations of π: $\pi_1 = 3.160494$ and $\pi_2 = 3.142857$. Express the values using three decimals and subtract:

	exact value	approximate	abs. error	rel. error
π_1	3.160494	3.160	-0.00049	-0.00016
π_2	3.142857	3.143	0.00014	0.00005
$\pi_1 - \pi_2$	0.017637	0.017	-0.00064	-0.03612

The relative error of the difference is much bigger than the original errors, because the most significant digits of the mantissas partly cancel each others leaving a smaller number of significant digits (catastrophic cancellation).

Beware subtraction of nearly equal values!

For example, when x is small, the following expression may give problems:

$$\sqrt{1+x} - 1,$$

If $x << 1$, the square root can be replaced with the first terms of its Taylor expansion:

$$\sqrt{1+x} - 1 \approx 1 + \frac{1}{2}x - 1 = \frac{1}{2}x.$$

The expression can also be converted to another form:

$$\sqrt{1+x} - 1 = \frac{x}{1 + \sqrt{1+x}}.$$

Example: find the integral

$$I_n = \int_0^1 \frac{x^n}{x+10} dx$$

$$\begin{aligned} I_{n+1} + 10I_n &= \int_0^1 \left(\frac{x^{n+1}}{x+10} + \frac{10x^n}{x+10} \right) \, dx \\ &= \int_0^1 \frac{x^n(x+10)}{x+10} \, dx \\ &= \int_0^1 x^n \, dx = \frac{1}{n+1}, \end{aligned}$$

which gives a recurrence relation

$$I_{n+1} = \frac{1}{n+1} - 10I_n.$$

$$I_0 = \int_0^1 \frac{dx}{x+10} = \Big|_0^1 \ln(x+10) = \ln 11 - \ln 10 \approx 0.0953.$$

$$\begin{aligned}
I_1 &= \frac{1}{1} - 10 \times 0.0953 \approx 0.0470, \\
I_2 &= \frac{1}{2} - 10 \times 0.0470 \approx 0.0300, \\
I_3 &= \frac{1}{3} - 10 \times 0.0300 \approx 0.0333, \\
I_4 &= \frac{1}{4} - 10 \times 0.0333 \approx -0.0833.????
\end{aligned}$$

The denominator has almost a constant value 10; hence

$$I_n \approx \frac{1}{10} \int_0^1 x^n \, dx = \frac{1}{10} \frac{1}{n+1}.$$

Thus the problem is the subtraction of nearly equal quantities.