1.2 Error propagation

When solving problems (mathematical models) on computers, there are at least three sources of errors:

- *Data errors*: For example, input data (constants and parameters) are given from physical measurements, and are therefore subject to measurement errors.
- *Rounding errors*: Only a finite set of numbers can be represented in a computer, and each step in a sequence of operations produces a rounding error.
- *Approximation errors*: An approximation to the solution of the problem is usually found by some numerical method.

If x is the exact value, the approximated value is given by

$$\tilde{x} = x + \Delta x = x(1 + \delta x)$$

where Δx is the absolute error and $\delta x = \Delta x / x$ the relative error.

We will now see how errors in input data, represented by $x = (x_1, x_2, \ldots, x_m) \in D \in \mathbb{R}^m$ propagates in the model given by

$$y = \varphi(x), \qquad \varphi : D \to \mathbb{R},$$

where y is the output of the model. We will assume φ to be sufficiently differentiable. Given errors in the input data x_i, we get

$$y + \Delta y = \varphi(x_1 + \Delta x_1, \ldots, x_m + \Delta x_m) = \varphi(x) + \sum_{i=1}^{m} \frac{\partial \varphi}{\partial x_i} \Delta x_i$$

Here quadratic terms are ignored. Thus, the absolute error in y is given by

$$\Delta y = \sum_{i=1}^{m} \frac{\partial \varphi}{\partial x_i} \Delta x_i$$

and the relative error

$$\delta y = \frac{\Delta y}{y} = \sum_{i=1}^{m} \frac{\partial \varphi}{\partial x_i} \frac{x_i}{\varphi} \, \delta x_i.$$

From the triangle inequality, we get

$$|\delta y| \leq \sum_{i=1}^{m} \left| \frac{\partial \varphi}{\partial x_i} \frac{x_i}{\varphi} \right| \, |\delta x_i|.$$

The expression $\left| \frac{\partial \varphi}{\partial x_i} \frac{x_i}{\varphi} \right|$ is thus the factor of which the error in x_i is amplified or damped, and is often referred to as the *condition numbers* of the problem. If the condition number is $\gg 1$ the problem is ill-conditioned, otherwise it is well-conditioned. If the number is > 1, the error is amplified, so if φ is a part of a process and will be repeated several times, the complete process will be unstable.

From this, the error propagation of some common operations can be derived:

φ	Absolute error	Relative error
$y = x_1 + x_2,$	$\Delta y = \Delta x_1 + \Delta x_2,$	$\delta y = \dfrac{x_1}{x_1 + x_2}\delta x_1 + \dfrac{x_2}{x_1 + x_2}\delta x_2$
$y = x_1 x_2,$	$\Delta y = x_2 \Delta x_1 + x_1 \Delta x_2,$	$\delta y = \delta x_1 + \delta x_2,$
$y = \dfrac{x_1}{x_2},$	$\Delta y = \dfrac{1}{x_2}\Delta x_1 + \dfrac{x_1}{x_2^2}\Delta x_2,$	$\delta y = \delta x_1 - \delta x_2,$
$y = \sqrt{x},$	$\Delta y = \dfrac{1}{2\sqrt{x}}\Delta x$	$\delta y = \frac{1}{2}\delta x.$

With respect to the relative errors in y, we observe that addition is ill-conditioned if $x_1 \approx -x_2$, thus subtraction of two almost equal numbers should be avoided. All other operations in this table are well-conditioned. With respect to the absolute error, division with x_2 small compared to x_1 will amplify the errors, so will taking the square root of small numbers.

Example 1.1. *Given the quadratic equation*

$$x^2 + px + q = 0, \qquad p > 0$$

with solutions $x = (-p \pm \sqrt{p^2 - 4q})/2$. *Let us consider the computation of one of the roots,*

$$x_1 = \varphi(a, b) = \frac{-p + \sqrt{p^2 - 4q}}{2}.$$

The relative error in x_1 *is*

$$\delta x_1 = -\frac{p}{\sqrt{p^2 - 4q}}\,\delta q + \frac{p + \sqrt{p^2 - 4q}}{2\sqrt{p^2 - 4q}}\,\delta q.$$

If $p > 0$ *and* $q < 0$ *then the condition numbers satisfy*

$$\left| -\frac{p}{\sqrt{p^2 - 4q}} \right| < 1 \qquad \text{and} \qquad \left| \frac{p + \sqrt{p^2 - 4q}}{2\sqrt{p^2 - 4q}} \right| < 1.$$

In this case, the problem is really well-conditioned. But if $p^2 \approx 4q$ *the condition numbers can be large and the problem ill-conditioned. These conclusions also hold for the second root (check it yourself).*

What about the practical computations. Let us assume the well-conditioned case, $p > 0$ *and* $q < 0$. *In the computer, the following computations will be performed to compute the two roots:*

$$\begin{aligned}
r &= p^2 \\
s &= r - 4q \\
t &= \sqrt{s} \\
u_1 &= -p + t \qquad\qquad & u_2 &= -p - t \\
x_1 &= u_1/2 & x_2 &= u_2/2
\end{aligned}$$

According to the discussion above, all operations are harmless, except for possibly the computation of u_1. *If* $p^2 \gg -4q$ *then* $p \approx t$ *and we have subtraction of two almost equal numbers. We can illustrate this numerically by an example: Let* $p = 1.2$ *and* $q = -1.4 \cdot 10^{-8}$. *The roots* $\tilde{x}_i$ *are found by the straightforward operation in matlab* `x1 = (-p+sqrt(p^2-4*q))/2` *and similar for* x_2. *The result, together with the exact values of the roots are*

$$\begin{aligned} x_1 &= 1.166666655324074\ldots \cdot 10^{-8}, & \tilde{x}_1 &= 1.166666652174797 \cdot 10^{-8}, & |\delta x_1| &= 2.7 \cdot 10^{-9}, \\ x_2 &= -1.200000011666666\ldots, & \tilde{x}_2 &= -1.200000011666666, & |\delta x_2| &\sim \varepsilon, \end{aligned}$$

where $\varepsilon = 2.2 \cdot 10^{-16}$ *is the machine precision. The error in* x_1 *may still seem small, but it has in fact been amplified by a factor of approximately* 10^7. *In this case, there is a simple remedy. Noticing that* $x_1 x_2 = q$ *makes it possible to compute* $x_1 = q/x_2$, *which is a well conditioned operation. In fact, we get*

$$\tilde{x}_1 = 1.166666655324074 \cdot 10^{-8}, \qquad |\delta x_1| \sim \varepsilon.$$

To sum up:

- Condition numbers tell how much an error in input data can be amplified by the model.
- Rounding errors may cause mayhem even in well behaved-problems. Sometimes, but not always, the problem can be solved by rearranging the computations.
- Avoid subtraction of two almost equal numbers.