Error Analysis

If A is an approximation of an exact number E then

$$\Delta = E - A \qquad \text{true error}$$

$$|\Delta| = |E - A| \qquad \text{absolute error}$$

$$\varepsilon = \frac{|\Delta|}{|E|} \qquad \text{relative error.}$$

Sources of errors.

1) Input errors: Errors in data given to software
2) Procedural errors: Errors arising from approximation scheme of algorithm
3) Propagation error: Results from amplification of input error.
4) Computational error: Results from finite representation of real numbers

Ill-conditioned algorithm: An algorithm with significant propagation error.

Unstable algorithm: Significant computational error.

Estimating propagation error

Consider the computation

$$y = f(x_1, x_2, x_3, \ldots, x_n)$$

and let $e_1, e_2, \ldots, e_n$ be the input errors relative errors of $x_1, x_2, \ldots, x_n$. The error e of y is estimated by:

$$e = \frac{\Delta y}{y} = \sum_{a=1}^{n} \frac{x_a}{f(\vec{x})} \frac{\partial f(\vec{x})}{\partial x_a} e_a$$

- $f(x_1, x_2) = x_1 x_2 \Rightarrow e = e_1 + e_2$
- $f(x_1, x_2) = x_1 / x_2 \Rightarrow e = e_1 - e_2$
- $f(x_1, x_2) = x_1 + x_2 \Rightarrow e = \dfrac{x_1 e_1 + x_2 e_2}{x_1 + x_2}$

  For $x_1 \geq 0$ and $x_2 \geq 0 \Rightarrow e \leq e_1 + e_2$

} Well conditioned

- $f(x_1, x_2) = x_1 - x_2 \Rightarrow e = \dfrac{x_1 e_1 - x_2 e_2}{x_1 - x_2}$

For $x_1 \approx x_2 \rightarrow e$ is amplified by denominator!

Thus it is not safe to subtract numbers that are nearly equal to each other.
↳ Eliminate/Reduce subtractions.

- $f(x_1) = x_1^p \Rightarrow e = pe_1$

Thus, roots ($0<p<1$) are well conditioned but powers with large $p \gg 1$ are ill-conditioned

<u>example 1</u> : Quadratic formula $ax^2+bx+c=0$
Assume $b>0$.
For numerical accuracy rewrite

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{as:} \quad \begin{cases} x_1 = \dfrac{-b-\sqrt{\Delta}}{2a} \\ x_2 = \dfrac{2ac}{-b-\sqrt{\Delta}} \end{cases}$$

<u>example 2</u> : Evaluation of $p(x) = a_n x^n + \cdots + a_1 x + a_0$.

We use the Horner scheme.
$p(x) = (\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots)x + a_0$
which can be written recursively as:

$$\begin{cases} b_n = a_n \\ b_k = b_{k+1} x + a_k \text{ , for } k = n-1, \dots, 0 \end{cases}$$

$$\Downarrow$$

$$p(x) = b_0.$$

The Horner scheme alternates addition with multiplication.

Understanding numerical stability.

Consider the iteration

$$\begin{cases} x_0 = 1 \text{ , } x_1 = 1/3 \\ x_{n+1} = \frac{13}{3} x_n - \frac{4}{3} x_{n-1} \end{cases} \quad (1)$$

By induction, we can easily prove that

$$x_n = (1/3)^n.$$

A numerical evaluation of this iteration diverges. The origin of this problem is the "unstable" mode in the general solution of (1)

$$x_n = A(1/3)^n + B \cdot (4)^n.$$