MATH 126 Lecture 3

Catastrophic cancellation: loss of significant digits that results from substracting nearly equal numbers

Example
$$\begin{array}{r} 123.4567 \\ -\ 123.4566 \\ \hline 000.0001 \end{array}$$

123.4567 → seven-digit accuracy

000.0001 → 1-digit accuracy

Loss of significance

In the above, it is quite evident that there is loss of significance. In general, it might require effort to know where cancellation happens

Strategies

A. Increase precision

B. Restructure the formula/expression so that it is not prone to cancellation

We prefer approach B and illustrate this via several examples.

Example: calculate $\sqrt{9.01} - 3$
(Assume we are using a 3-decimal digit arithmetic)

Solution: $\sqrt{9.01} \approx 3.001662 \Longrightarrow \boxed{3.00}$ in 3-digits

$\sqrt{9.01} - 3 = 0$ in 3-digits.

What can we do?

$$\sqrt{9.01} - 3 = \sqrt{9.01} - 3 \cdot \frac{(\sqrt{9.01} + 3)}{(\sqrt{9.01} + 3)}$$

$$= \frac{9.01 - 3^2}{\sqrt{9.01} + 3} = \frac{0.01}{6} = 0.00167 \approx 1.67 \times 10^{-3}$$

correct answer $= 1.6682 \times 10^{-3}$

Example   Solve $ax^2 + bx + c = 0$
with $a = 1$, $b = 68.50$ and $c = 0.1$

* use base-10 arithmetic with 4 significant digits

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x_1 = \frac{-68.50 + \sqrt{4692}}{2} = \frac{-68.50 + 68.50}{2} = 0$$

$$x_2 = \frac{-68.50 - \sqrt{4692}}{2} = \frac{-68.50 - 68.50}{2} = -68.50$$

However, the correct roots are $X_1 = -0.001460$
$X_2 = -68.50$

Relative error in computing x_1 = 1 (Really bad!)

Question   How can we avoid this?

The two roots of a quadratic equation are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

$$x_1 \cdot x_2 = \frac{b^2 - (b^2 - 4ac)}{(2a)^2} = \frac{4ac}{4a^2} = \frac{c}{a}$$

When $4|ac| << b^2$, b and $\sqrt{b^2 - 4ac}$ are nearly equal leading to catastrophic cancellation

Solution $x_1 = \dfrac{-b + \text{sign}(b)\sqrt{b^2 - 4ac}}{2a}$

$x_2 = \dfrac{c}{ax_1}$

Example: suppose we need to compute the integral $E_n = \int_0^1 x^n e^{x-1}\,dx$ for different values of n.

Solution
$$E_n = \int_0^1 x^n e^{x-1}\,dx$$
$$= x^n e^{x-1}\Big|_0^1 - \int_0^1 n x^{n-1} e^{x-1}\,dx$$
$$= 1 - n E_{n-1} \qquad n = 2, 3, \ldots$$

If we know E_1, we can compute the other E_n recursively

Let's use six digits approximation

$$E_1 = \int_0^1 x e^{x-1}\,dx = x e^{x-1}\Big|_0^1 - \int_0^1 e^{x-1}\,dx$$
$$= 1 - \left[e^{x-1}\Big|_0^1 \right]$$
$$= 1 - [1 - e^{-1}] = 1/e \approx 0.367879$$

Let's see the other values.

$$E_2 = 0.264242$$
$$E_3 = 0.207274$$
$$E_4 = 0.170904$$
$$\vdots$$
$$E_9 = -0.068480$$

Why is E_9 wrong?
$\downarrow$
Integrand is positive in the interval $[0, 1]$

Let's understand what happened.

$$E_1^{true} = E_1 + \text{Error} \qquad \text{Error} \approx 4.412 \times 10^{-7}$$
$$= E_1 + \varepsilon$$
$$E_2 = 1 - 2E_1 = 1 - 2(E_1^{true} - \varepsilon)$$
$$= (1 - 2E_1^{true}) + 2\varepsilon$$
$$= E_2^{true} + 2\varepsilon$$
$$E_3 = 1 - 3E_2 = 1 - 3(E_2^{true} + 2\varepsilon)$$
$$= E_3^{true} + 6\varepsilon$$

Error gets magnified at each step

$$E_9 = 9! \times 4.412\times10^{-7} = 0.1601$$

Problem: How can we fix this error?
Rewrite the recurrence relation.

$$E_{n-1} = \frac{1-E_n}{n} \quad n = \ldots, 3, 2$$

However, we need a starting value

$$E_n = \int_0^1 x^n e^{x-1}\,dx \le \int_0^1 x^n\,dx = \left.\frac{x^{n+1}}{n+1}\right|_0^1 = \frac{1}{n+1}$$

choose large n such that $E_n \approx 0$

Example: $n = 20$, $E_{20} \le \frac{1}{21}$

set $E_{20} = 0$ and apply recurrence backwards

$E_9 = 0.0916123$ (correct to six-digit precision)

Backward Stability: An algorithm is called backward stable if it produces an exact solution to a nearby problem

* This idea is due to J.H. Wilkinson (1919-1986)

Notation: $V \equiv$ set of input data

We think of our algorithm as a function $f(x)$ where $x \in V$

Formal definition: An algorithm f is called backward stable if for every $x \in V$, there is $\tilde{x} \in V$ with $f(\tilde{x}) = \tilde{f}(x)$ and

$$\frac{\|\tilde{x} - x\|}{\|x\|} = O(\varepsilon) \qquad \varepsilon \equiv \text{machine precision}$$

Example: $x \longrightarrow fl(x)$ $\qquad fl(x) = x(1+\varepsilon)$

We need to show: $f(\tilde{x}) = \tilde{f}(x)$ $\qquad \tilde{x} = x(1+\varepsilon)$

$\downarrow$

$\tilde{x}$

Note $\frac{|\tilde{x} - x|}{|x|} = \varepsilon$

$\therefore$ backwards stable

Example $(x, y) \to fl(x+y)$

$f(\tilde{x}, \tilde{y}) = \tilde{x} + \tilde{y}$

$$\begin{aligned}\tilde{f}(x, y) &= fl(x+y) \\ &= (x+y)(1+\delta) \\ &\text{where } |\delta| \le \varepsilon_{machine} \\ &= x(1+\delta) + y(1+\delta) \\ &= \tilde{x} + \tilde{y}\end{aligned}$$

Note that $\frac{|x-\tilde{x}|}{|x|} = O(\varepsilon_{mach})$

$\frac{|y-\tilde{y}|}{|y|} = O(\varepsilon_{mach})$

Example $(x) \to x+1$

$\tilde{f}(x) = (1+x)(1+\delta) \qquad |\delta| \le \varepsilon_{machine}$

$\tilde{f}(x) = 1 + \underbrace{\delta + x + x\delta}_{x\left(1 + \frac{\delta}{x} + \delta\right)}$

$\frac{\left|x\left(1+\frac{\delta}{x}+\delta\right) - x\right|}{|x|} \le \left|\frac{\delta}{x} + \delta\right|$ Large relative error near $x \approx 0$

Not always a reasonable notion of stability. However, for most algorithms in numerical linear algebra, we can show that we can achieve backword stability.

Conditioning

Small perturbations of input data $\xrightarrow[\text{conditioned}]{\text{well}}$ small perturbations of output

small perturbations of input data $\xrightarrow[\text{conditioned}]{\text{ill}}$ Large perturbations of output

If $f: \mathbb{R}^n \to \mathbb{R}^m$ and x and $y \in \mathbb{R}^n$, then the condition number is defined as

$$\lim_{\epsilon \to 0} \sup \frac{\|f(y) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|x-y\|}{\|x\|} \quad \text{s.t. } \|x-y\| \le \varepsilon$$

For differentiable functions, it can be shown that $\kappa(x) = \frac{|f'(x)|}{|f(x)|} |x|$

Example $f(x) = \sqrt{x}$

$f'(x) = \frac{1}{2} x^{-1/2}$

$\kappa(x) = \frac{|f'(x)|}{|f(x)|} |x| = \frac{1}{2} \frac{x}{\sqrt{x}\sqrt{x}} = 1/2$

well-conditioned problem

Example $f(x) = e^{x^2}$ $x \in \mathbb{R}$

$f'(x) = 2x e^{x^2}$

$\kappa(x) = \frac{|f'(x)|}{|f(x)|} |x|$

$= 2x^2$

For small x, well-conditioned

For large x, ill-conditioned

Important takeaway $\equiv$ Forward error $=$ stability $+$ conditioning