UCLA Math151A Fall 2021
Lecture 2
20210927

Errors and Convergence Rates

Optional reading: book 1.2, 1.3, 2.4

We should be aware of some facts:

- Computers have finite memeory and stoerage. **But real numbers can have an infinite number of digits, e.g. π, e, ... They cannot be represented exactly using floating point numbers in computer.**

- Only a subset of the rational numbers Q can be represented exactly

- Working with floating point numbers instead of real numbers produces round off error. How to quantify it?

Errors

Definition 2.1. Let $p \in \mathbb{R}$, and $\tilde{p}$ be an approxiamtion to p , then the **absolute error:**

$$e_{abs} := |p - \tilde{p}|. \tag{2.1}$$

(:= means define) □

Definition 2.2. The **relative error:**

$$e_{rel} := |\frac{p - \tilde{p}}{p}|. \tag{2.2}$$

□

Note: not defined when $p = 0$, in which case typically just use aboslute error.

Note: the percent error is simply $e_{rel} \cdot 100\%$.

Example 2.1. $p = 1$, $\tilde{p} = 0.9$, or $p = 1000, \tilde{p} = 900$, then for the first case,

$$e_{abs} = 0.1$$
$$e_{rel} = 0.1$$

For the second case

$$e_{abs} = 100$$
$$e_{rel} = 0.1.$$

The relative error is thus usually a more useful measure. □

Finite Digit Arithmetic

For example, numbers like π has to be truncated, Computers cannot store them all. Because computers have finite memory, we can either **chop** or **round** numbers that cannot be represented exactly.

Example 2.2. Bar is for infinite recurring decimal. $x = \frac{5}{7} = 0.\overline{714285}$, $y = \frac{1}{3} = 0.\overline{3}$. Suppose we use 5 digit arithmetic.

Let $fl(x)$ denote the floating point approx to x.

With **chopping**:

$$fl(x) = 0.71428$$
$$fl(y) = 0.33333$$
$$fl(fl(x) + fl(y)) = fl(1.04761) = 1.0476$$

If it used **rounding**, then $fl(x) = 0.71429$. □

Note that 5 digit meaning counting significant digits to 5.

Remark 2.1. In reality things are in binary. **This class focus on decimal for simplicity.** □

Example 2.3. Consider 3 digit rounding arithmetic.

$$z = 340.72$$
$$fl(z) = 341$$

Example 2.4. Let $f(x) = x^3 - 6.1x^2 + 3.2x + 1.5$. Let $x = 4.71$,

$$f(4.71) = -14.263899,$$

thsi is the exact answer. With 3 digit roudning arithmetic

$$fl(x^2) = fl(4.71 * 4.71) = fl(22.1841) = 22.2$$

$$fl(x^3) = fl(x * fl(x^2)) = fl(4.71 * 22.2) = fl(104.562) = 105$$

$$fl(6.1x^2) = fl(6.1fl(x^2)) = fl(6.1 * 22.2) = fl(135.42) = 135$$

$$fl(3.2x) = fl(15.072) = 15.1$$

$$fl(f(x)) = fl(105 - 135 + 15.1 + 1.5) = -13.4$$

$$fl(f(4.71)) = -13.4$$

which is $\approx 6\%$ error. How to reduce it? □

Remark 2.2. Every operation, multiply, add, introduces error in the process. □

Example 2.5. Revisiting the above example: In "naive" implmentation, there are 7 FLOPS. (A F.L.O.P = floating point operation) But we can improve the FP error by **nesting** our calculation.

Analytically:

$$\begin{aligned} f(x) &= x^3 - 6.1x^2 + 3.2x + 1.5 \\ &= ((x - 6.1)x + 3.2)x + 1.5 \text{ (written in nested form)} \end{aligned}$$

Doing calculutaion of $fl(f(4.71))$ requires only 5 FLOPs with the nested form.

Using nested form, the rel error is about 0.25%!

$$fl(f(4.71)) = -14.3$$

Takeaway: It's best to reduce the FLOPs whereever possible.

Order of Convergence for Sequences

Definition 2.3 (Convergence order of **convergent** sequences)**.** Let $(p_n)_{n\in\mathbb{N}}$ ($\mathbb{N}$ is natural numbers)

$$= (p_1, p_2, p_3, \dots)$$

be a **convergent** sequence in $\mathbb{R}$. Let $p_n \to p$ as $n \to \infty$. And assume $p_n \neq p$ for each n. Then, if $\exists \lambda, \alpha$ with $0 < \lambda < \infty$ (a finite and positive λ) and $\alpha > 0$ s.t.

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda,$$

then we say p_n converges to p with **order** α. □

Note that the denominator is not zero because we required that $p_n \neq p$.

Example 2.6. **suppose an iterative algorithm reduces the error between the estimate and the true solution by** 80% **per iteration, that is,** $p_1 = 1, p_2 = \frac{1}{5}, \ldots, p_n = \frac{1}{5}p_{n-1}$. Then $p_n \to 0$ as $n \to \infty$.

$$\frac{|p_{n+1} - 0|}{|p_n - 0|^1} = \frac{(1/5)^n}{(1/5)^{n-1}} = \frac{1}{5} = \lambda$$

So $\alpha = 1$. Therefore p_n converges with $\alpha = 1$ (linearly). □

Remark 2.3. (1) If $\alpha = 1$, (p_n) converges "linearly". Note that in this case (a general linear convergent case) $\lambda < 1$ is always true because otherwise the sequence is no way convergent (p_{n+1} will become further away from p). If $\alpha = 2$, (p_n) converges "quadratically", **there can be** $\lambda > 1$**, e.g. 1e-2, 1e-3, 1e-5, 1e-9, ...** $(|p_{n+1} - 0|/|p_n - 0|^2 = 10)$. If $\alpha = 3$, (p_n) converges "cubicly", etc. (2) $\alpha > 0$ is a real number (not necessarily a natural number). When $\alpha = 1.1$ we usually just say something like "converges with order 1.1". □

Practice: try to see whether the above example work for $\alpha = 2$. **(does not work as** $\lim_{n\to\infty} |p_{n+1} - 0|/|p_n - 0|^2$ **goes to** ∞**)**

Big O notation

In numerical analysis, sometimes we also like using the "big O" notation to describe how fast things change/converge to some limits.

Definition 2.4 (Big O notation)**.** $a(t) = O(b(t))$ with $b(t) > 0$ means "on the order of",

$$\exists\, C > 0 \text{ s.t. } |a(t)| \leq Cb(t) \text{ for } t \to 0 \text{ or } \infty$$

We mostly deal with $t \to 0$ case, and in such cases we often have $b(t) \to 0$. I.e., $a(t) = O(b(t))$ is equivalent to saying

$$\lim_{t \to 0} \frac{|a(t)|}{b(t)}$$

is bounded by a positive number. □