1 Preliminaries

- There is a pre-class assignment due before Class 02.
- There will be a skill check in class during Class 02. The problem info is below.
- Problem set 01 will be assigned Friday Sept 2 and is due Friday Sept 9 (content from classes 01 and 02).
- OH this week: today from 2:30-3:30pm in Pierce 316 (my office).

Big picture

Many of the math problems you have studied in the past (finding solutions to equations or systems of equations, approximating data by functions, finding maxima or minima of functions, computing derivatives or integrals, finding solutions to differential equations, approximating functions by other functions) can be tackled using computational methods. Typically these methods produce approximate solutions to the math problem.

During the semester, we will examine three factors in how approximate these solutions are: (1) the sensitivity of the output of a particular function to small changes in the input, (2) how numbers are represented in computers, and (3) the characteristics of the algorithms being used to compute a solution.

Today's class focuses on factors (1) and (2).

Skill check C01 practice

Let $f(x) = \cos x$. Find the condition number as a function of x. Use relative error to construct your expression.

Skill check solution

Using Taylor expansion to first order about x, $f(x + \Delta x) \approx f(x) + \Delta x \left.\frac{df}{dx}\right|_x$.

Working with a general $f(x)$: cond $\# = \left|\frac{\hat{y} - y}{y}\right| / \left|\frac{\hat{x} - x}{x}\right| \approx \left|\frac{\Delta x f'(x)}{f(x)}\right| / \frac{\Delta x}{x} = \left|x \frac{f'(x)}{f(x)}\right|$

For $f(x) = \cos x$, we have

cond $\# \approx \left|x \frac{-\sin x}{\cos x}\right| = |x \tan x|$

Teams

Not assigned: work with students sitting nearby.

---

2 Well conditioned vs ill conditioned problems

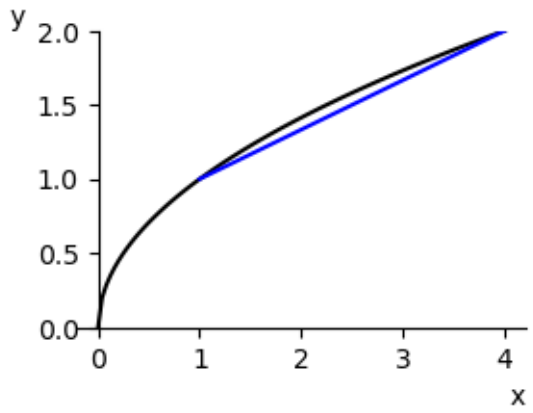
Notes based on work by Thomas Fai, APMTH 111 Spring 2017

Example: error from approximation

Evaluate $f(x) = \sqrt{x}$ (find the square root of some number).

Ex: Find $y = \sqrt{\pi}$. $\sqrt{\pi} \approx \sqrt{3}$. Note that $\sqrt{4} = 2$ and $\sqrt{1} = 1$.

Use a linear approximation for $\sqrt{3}$ to find an approximate value, $\hat{y}$ for $\sqrt{\pi}$.



For $x = \pi$: true solution is $y = 1.77245...$
The computed solution, $\hat{y}$, would be the true solution with a different input, $\hat{x}$.

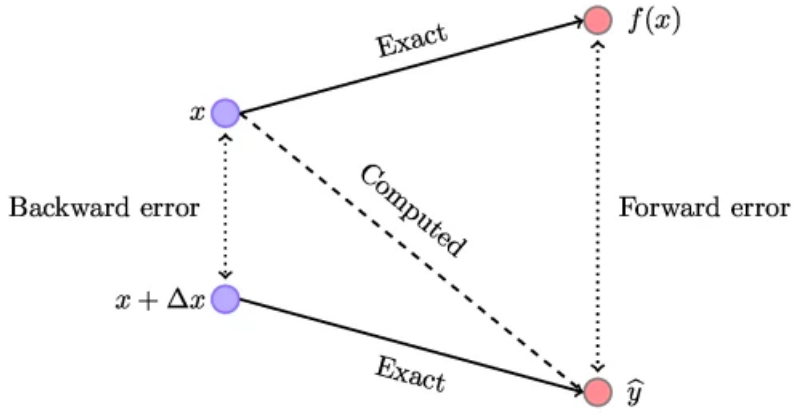


Image from Nick Higham's blog

Definitions: error

- The **forward error** is given by $|y - \hat{y}|$ (**absolute**) or $\left|\frac{y - \hat{y}}{y}\right|$ (**relative**).
- The **backward error** is given by $|x - \hat{x}|$ (**absolute**) or $\left|\frac{x - \hat{x}}{x}\right|$ (**relative**).
- The **condition number** is a measure of the sensitivity of the output to small perturbations in the input [1]. It is given by the ratio of forward to backward error: cond $\# = \frac{|(y - \hat{y})/y|}{|(x - \hat{x})/x|}$.
- If either x or y is close to zero then the condition number may be calculated using absolute error: cond $\# = \frac{|(y - \hat{y})|}{|(x - \hat{x})|}$

Condition number

Consider $y = f(x)$ for some function f. If the condition number is ≈ 1, and you change your input value by 1%, how do you expect the output value to change?

What if the condition number is 100? In this case, if you change your input value by 1%, how do you expect the output value to change?

- A problem is **well conditioned** when small changes in the input lead to comparably small changes in the output.
- A problem is **ill conditioned** when the output is sensitive to small changes in the input (or the reverse).

Condition number examples

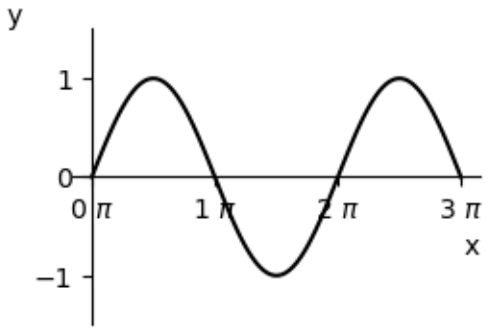
Set $\hat{x} = x + \Delta x$, so Δx is the backward error. Use linear approximation / Taylor expansion to find the condition number (as a function of x) for the following functions.

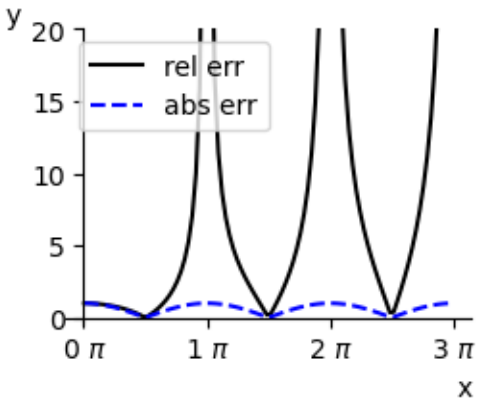
Recall: to first order (a linear approximation), $f(\hat{x}) \approx f(x) + (\hat{x} - x)f'(x)$

1. Let $f(x) = ax + b$. Find the condition number.

2. Let $f(x) = e^x$. Find the condition number.

3. Let $f(x) = \sin x$. Find the condition number.





Sources of error in solving problems computationally

1. The problem is ill-conditioned, so will be sensitive to small changes in the input (cond $\# \gg 1$ or to small changes in the output (cond $\# \ll 1$). This is a feature of the problem itself (not the methods used to tackle the problem).
2. Instability of the solution method. A stable algorithm gives exactly the right answer to a nearby problem (i.e. small backward error).

3 Representing numbers in a computer (floating point)

In a computer, we assign finite memory to store each number.

- Only some numbers can be represented: there is no continuity. Every number has a neighborhood about it with no other numbers.
- Numbers for which the representation would be infinite are always shortened (ex: π, $\sqrt{2}$, $0.\bar{3}$).

- A **floating point number** consists of three parts: the **sign** of the number, the string of bits (**mantissa**), and an **exponent**. [2]
- These parts are stored together.
- A floating point system can be characterized by four numbers:
  - β (base)
  - p (precision)
  - $[L, U]$ (range for exponents)
- A floating point number is of the form

$$\pm \left(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + ... + d_{p-1}\beta^{-(p-1)}\right)\beta^E$$

  where $0 \leq d_i \leq \beta - 1$ and $L \leq E \leq U$ are integers.

  In base 10 you would write $d_0.d_1d_2...d_{p-1} \times 10^E$.
- $\pm$ is the sign, $d_0d_1d_2...d_{p-1}$ is the mantissa, E is the exponent, and p is the precision.
- In a **normalized** system we require $d_0 \neq 0$.

Example

Consider a normalized floating point system with $\beta = 10$, $p = 4$, $-5 \leq E \leq 5$.

- For numbers in this system, find the smallest number ϵ such that $1 + \epsilon > 1$. This number is referred to as machine epsilon, ϵ_{mach}

- How would you represent π in this system? *If there are multiple options, explain your choice.*

- Find the smallest number represented in the system. This is the underflow level (UFL).

- Find the largest number represented in the system. This is the overflow level (OFL).

- Compute $1.0 \times 10^{-5} + 1$ within this system.

Note: $1.0 \times 10^{-5} + 1.0 \times 10^{-5} = 2 \times 10^{-5}$
but $(1.0 \times 10^{-5} + 1) + (1.0 \times 10^{-5} - 1) = 0$. This is called **catastrophic cancellation**.

Rounding

- Let $\mathsf{fl}(x)$ denote the floating point approximation to x.
- To convert from x to $\mathsf{fl}(x)$ one option is to **chop**, taking the first p digits.

  Dropping the $p+1$st digit leads to a relative error of $\left|\dfrac{\mathsf{fl}(x) - x}{x}\right| \leq \beta^{-(p-1)}$
- Another option is **rounding to nearest** by taking the floating point number nearest to x.

  This leads to a relative error of $\left|\dfrac{\mathsf{fl}(x) - x}{x}\right| \leq \frac{1}{2}\beta^{-(p-1)}$

  Note: If x is equidistant between two floating point numbers (think of 1.5 when rounding to the nearest integer), then the value of d_{p-1} is used to determine whether to round up or down.
- The maximum relative error gives the accuracy of the floating point system. It is

$$\epsilon_{\mathsf{mach}} = \max_{x \in [\mathsf{UFL}, \mathsf{OFL}]} \left|\frac{\mathsf{fl}(x) - x}{x}\right|.$$

Example.
Let $\beta = 10$, $p = 2$.

	chop	nearest
2.344		
2.351		
2.389		
2.350		

Binary

- The floating point systems used in computers typically have $\beta = 2$ (binary), so the digits of the floating point numbers, $d_i = 0$ or 1. We will use b_i interchangeably with d_i when we are working in binary.
- A single binary digit is called a **bit**.
- In double precision, a floating point number has 64 bits.

$$se_1e_2...e_{11}b_1b_2...b_{52}$$

where s is the sign, followed by 11 bits for the exponent, and then the 52 bits following the decimal point (also called the **binary point** or **radix point**).
- A few exceptional numbers also need to be represented: NaN (not a number, i.e. $0/0$), Inf (infinity, i.e. $1/0$), 0 (note that $d_0 = 1$ so 0 needs a special representation)

References

[1] Nick Higham. What is a condition number? `https://nhigham.com/2020/03/19/what-is-a-condition-number/`, Mar 2020.

[2] T. Sauer. *Numerical Analysis*. Pearson Education, 2018.