LECTURE 1

- Computations involving $\boxed{\text{continuous}}$ mathematics
  - math arising in science & engineering
  - e.g., simulation
    financial models
    optimization problems
  - design + analysis of algorithms

historically: "numerical analysis"
- eg, vision, data mining, bio, ...

- ~~most~~ problems can't be solved $\boxed{\text{exactly}}$
  - find some iterative process that converges to the solution + cut it off at some finite pt. truncate
  - → seek rapid convergence
  - → assess error

* sometimes prefer iterative over direct

Simulation §1.1.1
- solve problems that can't otherwise be solved
- explore parameter space more economically
  no "build-and-test"
  "virtual prototyping"

Scientific Computing.

1. "mathematical model" – Applied Math
   - equations
2. algorithms to solve equations numerically.
3. implement
4. run
5. visualize or otherwise represent the result.
6. interpret & validate.

Well-Posedness (vs. ill-posedness)
- solution exists
- unique
- depends continuously on data ⊗

⊗ in numerical computations, small changes inevitable.
problem may be well-posed but still sensitive
- measure of sensitivity of a problem = "condition number"

stable algorithm - doesn't introduce sensitivities or ill-posedness to a well-posed problem.
(just because math. prob. well-posed, doesn't mean algorithm is ...). NaN's

1.1.2. General Strategy

infinite	← finite
dim spaces	dim spaces.
integrals	sums
derivatives	differences.
diff. eq.	algebraic eq.
nonlinear	linear
high order	low order
complicated functions	simple functions e.g. poly.
general matrices	simpler matrices

Approximations - w/ (arbitrarily) good accuracy

Sources of Error

1. Modeling

   simplify the problem to facilitate
   studying it or some aspect of it.

2. Empirical Measurements

   instrument error
   human error
   system noise

3. Other errors in input data

During computation ...

4. truncation error or discretization error

   turning infinite into finite
   cut off at some point or
   sample functions with limited accuracy.

5. Rounding

   represent real #'s w/ finite precision

Ex. Approx. area of earth

$A = 4\pi r^2$

↑ modeling error

$\approx 509{,}904{,}355.08376$

- 1 earth modeled as sphere
- 2 $r \approx 6370$ km approximate
- 4 $\pi \approx 3.1415926\ldots$
- 5 values are rounded in calculation

1.2.2 Absolute Error + Relative Error

- absolute error = approx. − exact

- relative error = $\dfrac{\text{absolute error}}{\text{exact}}$ *

  * in practice use the approximate value.

- Relative error can also be expressed as percentage
- Another useful interpretation: # of correct significant digits

  rel. err $\approx 10^{-p}$

  $\Rightarrow$ ~p significant digits. correct.

  $\frac{1}{1000} = 10^{-3}$ ✓

E.g. $\dfrac{100}{12345} \cong \dfrac{1}{100} = 10^{-2} \Rightarrow$ 2 sig. digits.

e.g. $\dfrac{\cancel{12345}}{12445}$ approx. soln

1.2.3 Data Error & Computational Error.

Compute $f(x)$, $f: \mathbb{R} \to \mathbb{R}$

x true input
$f(x)$ true output

$\hat{x}$ approx. input
$\hat{f}$ approx function evaluation

$$\text{total error} = \hat{f}(\hat{x}) - f(x)$$
$$= \hat{f}(\hat{x}) - f(x) + (f(\hat{x}) - f(\hat{x}))$$
$$= \underbrace{\hat{f}(\hat{x}) - f(\hat{x})}_{\text{computational error}} + \underbrace{\boxed{f(\hat{x}) - f(x)}}_{\text{propagated data error}} \ *$$

computational error: diff. of exact function & approx function on same input

propagated data error: diff. of exact function on approx. input & exact input.

* (choice of algorithm does not affect propagated data error.)

Ex. 1.2. Approximate $\sin(\frac{\pi}{8})$ $(\approx .3827)$

$\pi \approx \cancel{22/7}$ 3 $\qquad x = \pi/8$

$\hat{x} = \frac{3}{8}$

truncate

Taylor series $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$

$\sin x \approx x$

$$\sin\left(\frac{\pi}{8}\right) \underset{\substack{\uparrow \\ \text{propagated data error}}}{\approx} \sin\left(\frac{3}{8}\right) \underset{\substack{\uparrow \\ \text{computational error}}}{\approx} \frac{3}{8} = .375$$

$$\hat{f}(\hat{x}) - f(x) \approx .375 - .3827 = -.0077$$ total error

$$f(\hat{x}) = \sin(\tfrac{3}{8}) \cong .3663$$

propagated data error $\quad f(\hat{x}) - f(x) = .3663 - .3827 = -.0164$

computational error $\quad \hat{f}(\hat{x}) - f(\hat{x}) = .375 - .3663 = .0087$

Note: the two errors have opposite signs so partially offset each other, but could have same sign.

(Q) In what case above ~~w~~ would each type of error dominate?

$\boxed{1.2.8}$ Truncation Error & Rounding Error $\quad \boxed{\text{See Example}}$

- $\boxed{\text{truncation error}}$ — if we could use approx function w/ exact arithmetic, what would the result be

$f(x) - \hat{f}_{exact}(x)$

diff : = true result ~~w/ exact arith~~ — algorithm w/ exact arith. on actual data

E.g., • truncating infinite series
• derivatives ← finite diff.
• terminating iterative sequence before convergence.

$\hat{f}_{exact}(x) - \hat{f}_{round}(x)$ • $\boxed{\text{Rounding Error.}}$

diff between ~~algo~~ result produced using exact ~~arith~~ vs
& " " " finite-precision rounded arith

truncation error $= f(x) - \hat{f}_{exact}(x)$

rounding error $= \hat{f}_{exact}(x) - \hat{f}_{rounded}(x)$

$+$

$$f(x) - \hat{f}_{rounded}(x)$$

Ex. 1.3 Finite Difference Approximation

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Taylor's theorem

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \cdots$$

$$= f(x) + hf'(x) + \frac{h^2}{2} f''(\theta), \quad \theta \in [x, x+h]$$

truncation error:

$$f'(x) - \frac{f(x+h) - f(x)}{h} = \cancel{f'(x)} - \left(\cancel{f'(x)} + \frac{h}{2} f''(\theta)\right)$$

$$= -\frac{h}{2} f''(\theta) < M \frac{h}{2}$$

rounding error

rounding error in f $< \epsilon \Rightarrow$

$$\text{"}\frac{f(x+h) - f(x)}{h} - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} < \frac{1}{h}\left(\left|\tilde{f}(x+h) - f(x+h)\right| + \left|\tilde{f}(x) - f(x)\right|\right) \le \frac{2\varepsilon}{h}$$

total computational error

$$\frac{Mh}{2} + \frac{2\varepsilon}{h}$$

$h = 2\sqrt{\frac{\varepsilon}{M}}$

$$f(h) = \frac{M}{2} h + 2\varepsilon h^{-1} \Rightarrow f'(h) = \frac{M}{2} - 2\varepsilon h^{-2} = 0 \Rightarrow \frac{M}{2} = 2\varepsilon h^{-2}$$

$$h^2 = \frac{4\varepsilon}{M}$$

- all numerical values, input, intermediate, and output are rounded

----

Example: truncation vs. rounding error

- tradeoff between rounding error and truncation error when using finite-precision, floating-point arithmetic
- problem: computing the change in the surface area A of the Earth if its radius r ~= 6370 km changes by a given amount Δr. Two different formulas are used:
  - one from geometry, $\Delta A = 4 \pi (r + \Delta r)^2 - 4 \pi r^2$, that is theoretically exact (assuming perfect real arithmetic), and
    - for small Δr, large rounding error -> inaccurate
    - for large Δr, small rounding error -> accurate
  - the other a simple approximation derived from calculus, $\Delta A \approx 8 \pi r \Delta r$, whose accuracy depends on the amount by which the radius changes
    - for small Δr, small truncation error -> accurate
    - for large Δr, large truncation error -> inaccurate