Lecture Notes 07: Numerical Algorithms & Errors

CPSC 302: Numerical Computation for Algebraic Problems

Jessica Bosch
jbosch@cs.ubc.ca
http://www.cs.ubc.ca/~jbosch

University of British Columbia
Department of Computer Science

2017/2018 Winter Term 1

Outline

1. Waitlist
2. Roundoff Errors
   - Goal
   - Roundoff Error Propagation and Accumulation
3. Reflection

Outline

1. Waitlist
2. Roundoff Errors
3. Reflection

Waitlist Problem

SOLVED!!!

We are 81 students!

Outline

Goal

Identify & avoid different sources of roundoff error growth.

Roundoff Errors

- Roundoff error is generally inevitable in numerical algorithms involving real numbers.
- People often like to pretend they work with exact real numbers, ignoring roundoff errors, which may allow concentration on other algorithmic aspects.
- However, carelessness may lead to disaster!

  Example: Patriot missile failure in 1991

Roundoff Error Accumulation

- In general, if E_n is error after n elementary operations, cannot avoid linear roundoff error accumulation

$$E_n \simeq c_0 n E_0.$$

- Will not tolerate an **exponential** error growth such as

$$E_n \simeq c_1^n E_0 \text{ for some constant } c_1 > 1$$

  – an **unstable algorithm**.

- Example: recursive formula $y_n = \frac{1}{n} - 100 y_{n-1}$
  $\rightsquigarrow$ magnitude of roundoff errors gets multiplied by 100 each time

Roundoff Error Accumulation

When we design an algorithm, we must **keep error accumulation in mind**:

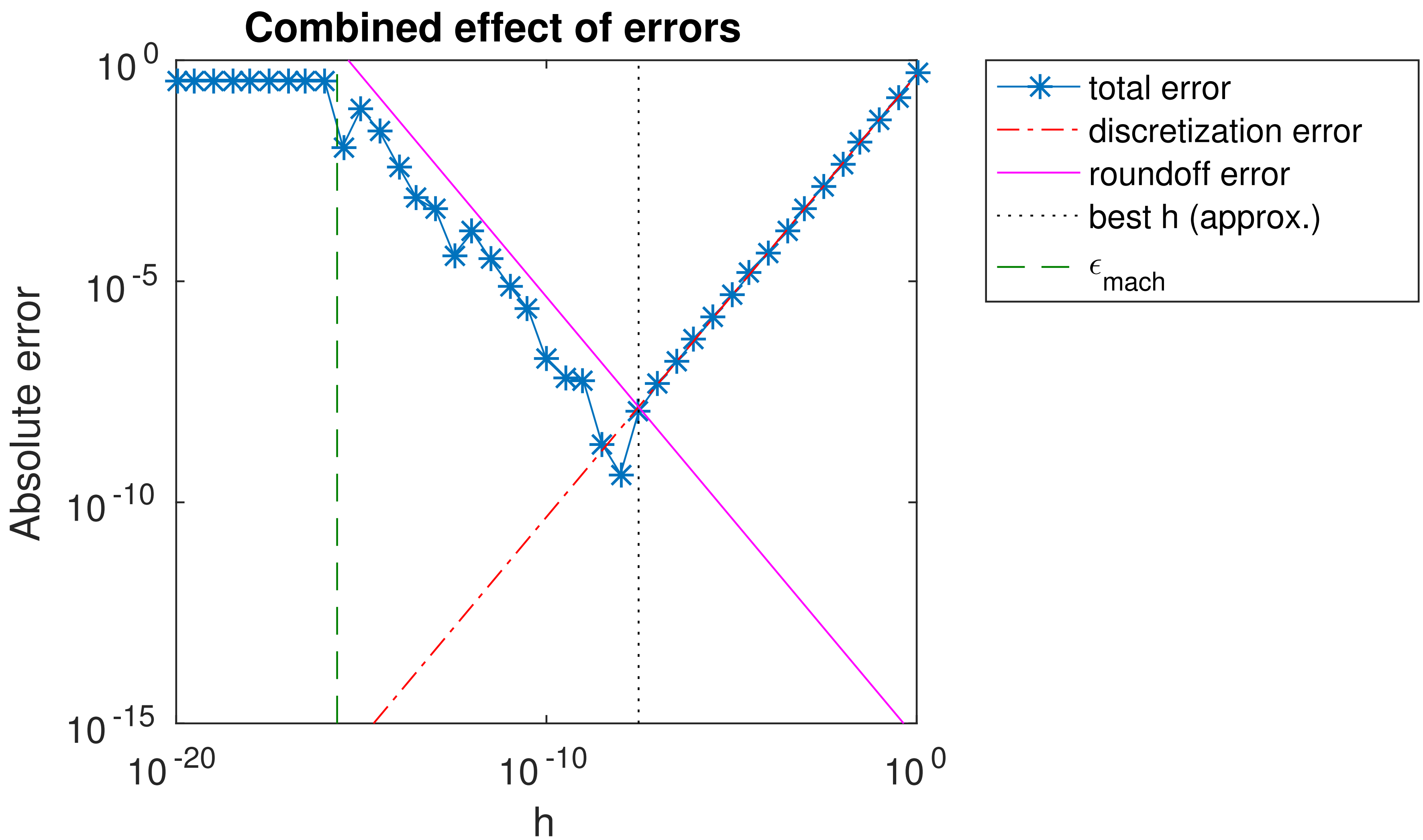
The computer may run millions of operations, and what may look like a meaningless error may quickly have a disastrous effect.

Cancellation Error

When two nearby numbers are subtracted, the relative error is large. That is, if $x \simeq y$, then $x - y$ has a large relative error.

This occurs in practice consistently and naturally, as we will see.

Review Example



$$\left| f'(x_0) - \frac{f(x_0+h)-f(x_0)}{h} \right|, \quad \frac{h}{2}\,|f''(x_0)|, \quad \frac{2\epsilon_{\text{mach}}}{h}, \quad 2\sqrt{\frac{\epsilon_{mach}}{|f''(x_0)|}}, \quad \epsilon_{mach} \approx 10^{-16}$$

Review Example

Function evaluation at nearby arguments:

- If $g(\cdot)$ is a smooth function then $g(t)$ and $g(t+h)$ are close for h small.
- But rounding errors in $g(t)$ and $g(t+h)$ are unrelated, so they can be of opposing signs!
- For numerical differentiation, e.g.

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}, \quad 0 < h \ll 1,$$

if the relative rounding error in the representation is bounded by ε_{mach} then in $|g(t+h) - g(t)|/h$ it is bounded by $2\,\varepsilon_{mach}/h$. This (tight) bound is much larger than ε_{mach} when h is small.

Another Example

Compute $y = \sinh(x) = \frac{1}{2}(e^x - e^{-x})$.

- Naively computing y at an x near 0 may result in a (meaningless) 0.
- Instead use Taylor's expansion

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots$$

to obtain

$$\sinh(x) = x + \frac{x^3}{6} + \dots.$$

- If x is near 0, can use $x + \frac{x^3}{6}$, or even just x, for an effective approximation to $\sinh(x)$.

So, a good library function would compute $\sinh(x)$ by the regular formula (using exponentials) for $|x|$ not very small, and by taking a term or two of the Taylor expansion for $|x|$ very small.

Another Example

Compute $y = \sqrt{x+1} - \sqrt{x}$ for $x = 1 \cdot 10^5$ in $\mathbb{F}$ with $t = 5$.

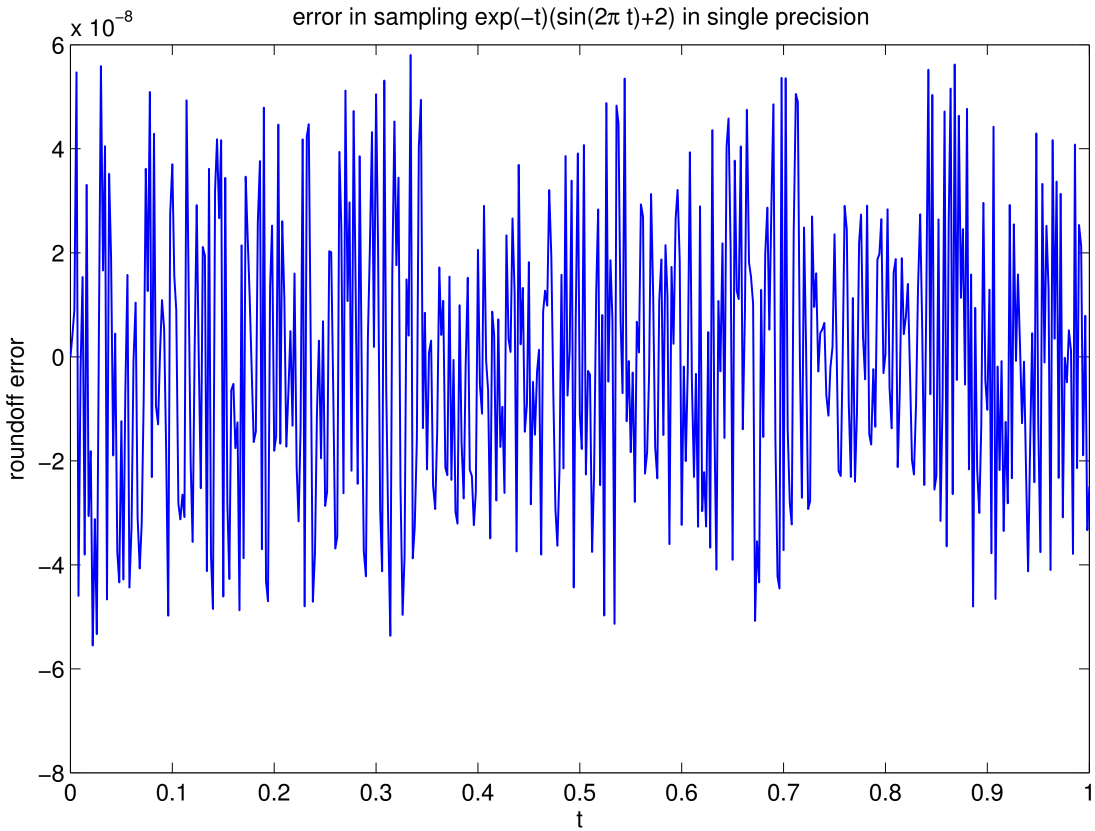
However, $x + 1 = 1.00001 \cdot 10^5 \notin \mathbb{F}$ and $\mathsf{fl}(x+1) = 1 \cdot 10^5 = x$. This results in $\mathbb{F}$ in $x + 1 = x$ and hence $y = 0$.

Instead use the identity

$$\sqrt{x+1} - \sqrt{x} = \frac{(\sqrt{x+1} - \sqrt{x})(\sqrt{x+1} + \sqrt{x})}{(\sqrt{x+1} + \sqrt{x})} = \frac{1}{\sqrt{x+1} + \sqrt{x}}.$$

The Rough Appearance of Roundoff Errors

Run program Example2_2Figure2_2.m



Note how the sign of the floating point representation error at nearby arguments t fluctuates as if randomly: as a function of t it is a "non-smooth" error.

Avoiding Overflow

Suppose $a \gg b$ and we wish to compute $c = \sqrt{a^2 + b^2}$.

Then it may be better to rescale and compute

$$c = a\sqrt{1 + (b/a)^2}.$$

This principle is actually used in the computation of the ℓ_2 norm of a vector:

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}.$$

Floating Point Arithmetic: Summary

- The order in which operations are performed matters.
  - Algebraically equivalent formulas may have very different roundoff errors, depending on their arguments.
  - For a single operation, roundoff error is usually small (around ϵ_{mach}), but poor formulas can make it **much** bigger in just a few operations (such as through cancellation).
- Algorithms using floating point will always be approximate, even if there is no discretization error.
- Algorithm design must keep error accumulation in mind.
  - Computers can run **a lot** of operations very fast.
  - Computational errors can compound and not just add up.

Outline

1. Waitlist

2. Roundoff Errors

3. Reflection

Reflection

On a piece of paper, answer the following question:

> *What was the muddiest point in the first course topic on "Numerical Algorithms & Errors"?*

Put the paper into the "Suggestion Box" or just on the table.