CSC 165

stability

week 12, lecture 1

Danny Heap

heap@cs.toronto.edu

www.cdf.toronto.edu/~heap/165/F09

resources: chapter 7 of course notes

http://docs.python.org/tutorial/floatingpoint.html

http://en.wikipedia.org/wiki/IEEE_754-2008

magnifying errors

cumulative error, $100 + 0.1 + 0.1 + \cdots$, and catastrophic cancellation, $11.1156 - 11.1264$, have a common feature: they magnify the rounding error

An algorithm, or expression, is called *unstable* if and only if errors in input get magnified in the output.

Sometimes you can reduce instability, for example by adding small quantities together first (example 1). It's not so easy to fix catastrophic cancellation — you could try increasing precision (t), which works until you get a pair of numbers even closer together. . .

one-time fixes

There's another vulnerability to catastrophic cancellation in the quadratic formula:

$$(x_1, x_2) = \left(\frac{-b + \sqrt{b^2 - 4ac}}{2a}, \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right)$$

Exactly one of the two roots is vulnerable to catastrophic cancellation, depending on whether $b > 0$ or $b \leq 0$. Solve the *other* one.

Suppose you have x_2. How do you compute x_1 without subtraction? Multiply the two roots.

in general. . .

Try to replace an unstable algorithm with a stable one (not always possible).

condition number reveals which functions may have a stable algorithm (next time)

Try increasing precision, which may reduce instability on some inputs.

penny piles

Start with 64 pennies on the left, 0 on the right.

By moving half an even pile to the other pile, can you achieve a pile with 40 pennies?