CSC 165
more floating-point
Danny Heap
heap@cs.toronto.edu
www.cdf.toronto.edu/~heap/165/W10

resources: chapter 7 of course notes
http://docs.python.org/tutorial/floatingpoint.html

SL06 - due Thurs.
office hours - continue
What's the condition number for $f(x) = x^5$? How about $f(x) = \cos(x)$?

What does this tell you about algorithms to implement $f$ in certain regions?

\[ \lim_{x \to 0} \frac{f(x)}{f'(x)} = \lim_{x \to 0} \frac{x^5}{5x^4} = 1 \]

\[ \|f(x) - f'(x)\| = \|x^5 - 5x^4\| = 0 \]

\[ \text{unbounded for } x \approx \frac{1}{2} \text{ near } x = 0 \]
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…

In a special case, such as the quadratic formula, you could solve for the root that avoids catastrophic cancellation, e.g. $x_1 = c/(x_2a)$, if you're faced with:

$$
(x_1, x_2) = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
$$
in general...

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

\[ \chi_1 = \frac{c}{\chi_2 \sigma} \]

*condition number* reveals which functions may have a stable algorithm

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

Many important functions are approximated by using part of their Taylor series expansion:

\[ e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \]

or \( \sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots \).

Calculus provides a bound on how much information you lose by truncation, and now you’ve got truncation and rounding as possible sources of error.
We want to apply the exact function, \( f \), to an exact value \( x \), yielding \( f(x) \).

We settle for an approximate function \( \hat{f} \) applied to an approximate value \( x' \), yielding \( \hat{f}(x') \).

We can break up the difference, \( |\hat{f}(x') - f(x)| \), into two parts, to account for the source of the error:

\[
|\hat{f}(x') - f(x)| 
\leq |\hat{f}(x') - f(x')| + |f(x') - f(x)|
\]