CS542G - Breadth in Scientific Computing

Web

- www.cs.ubc.ca/~rbridson/courses/542g
- Course schedule
    - Slides online, but you need to take notes too!
- Reading
    - There is an optional text, Heath
    - Relevant papers as we go
- Assignments + Final Exam information
    - Look for Assignment 1
- Resources

Contacting Me

- Robert Bridson
    - X663 (new wing of CS building)
    - Drop by, or make an appointment (safer)
    - 604-822-1993 (or just 21993)
    - email rbridson@cs.ubc.ca
- I always like feedback!
    - Ask questions if I go too fast…

Evaluation

- ~4 assignments (40%)
- Final exam (60%)

MATLAB

- Tutorial Sessions at UBC
- Aimed at students who have not previously used Matlab.
- Wed. Sept. 13, 9 - 10am, ICICS/CS x250.
- Wed. Sept. 13, 5 - 6pm, DMP 301.

  www.cs.ubc.ca/~mitchell/matlabResources.html

Units

- Floating Point
- Interpolation/approximation, dense linear algebra
- ODE's and time integration, tree methods
- Mesh generation, Poisson equation, sparse linear algebra
- Hyperbolic PDE's

Floating Point

Numbers

- Fixed Point
  - Can be very fast, but limited range - dangerous
- Arbitrary Precision Arithmetic
  - Tends to be very slow
  - Occasionally very useful in simple Extended Precision
- Interval Arithmetic
  - Track bounds on error with every operation
  - Slower
- Floating Point
  - Usually the best mix of speed and safety

Floating Point Basics

- Sign, Mantissa, Exponent
- Epsilon
- Rounding
- Absolute Error vs. Relative Error

IEEE Floating Point

- 32-bit and 64-bit versions defined
- Most modern hardware implements the standard
  - But Java gets it wrong
  - GPU's etc. often simplify for speed
- Designed to be as safe/accurate/controlled as possible

IEEE Special Numbers

- +/- infinity
  - When you divide 1/0 for example, or log(0)
  - Can handle some operations consistently
  - Instantly slows down your code
- NaN (Not a Number)
  - The result of an undefined operation e.g. 0/0
  - Any operation with a NaN gives a NaN
    - Clear traceable failure deemed better than silent "graceful" failure!
  - Nan != NaN

Cancellation

- The single biggest issue in fp arithmetic
- Example:
  - Exact arithmetic:
1.489106 - 1.488463 = 0.000643
  - 4 significant digits in operation:
1.489 - 1.488 = 0.001
  - Result only has one significant digit (if that)
- When close numbers are subtracted, significant digits cancel, left with bad relative error
- Absolute error is still fine…

Cancellation Example 1

- Can sometimes be easily cured
- For example, solving quadratic
  $ax^2+bx+c=0$
  with real roots

Cancellation Example 2

- Sometimes not obvious to cure
- Estimate the derivative an unknown function

Accumulation

- 2+eps=2
- (2+eps)+eps=2
- ((2+eps)+eps)+eps=2
- …
- Add any number of eps to 2, always get 2
- But if we add the eps first, then add to 2, we get a more accurate result

Exact numbers in fp

- Integers (up to the range of the mantissa) are exact
- Those integers times a power of two (up to the range of the exponent) are exact
- Other numbers are rounded
  - Simple fractions 1/3, 1/5, 0.1, etc.
  - Very large integers

Hardware

- Vectorization, ILP
- Separate fp / int pipelines
- Caches, prefetch
- Multi-processors

- Slow code:
- Fast code:
- Use good libraries when you can!