NUMERIC TESTING

Numeric functions

- Used for
  - Financial calculations
  - Real time control
    - Breaking distance
    - Flight control parameters
- Often depend on
  - Estimated and sampled data
  - Statistics

Testing numeric functions

- Two basic categories
  - Integer functions
  - Real number functions
- Discrete integer functions
  - Large but finite space
  - In theory possible to test every case
- Floating point inputs
  - Infinite domain mapped to finite address space

Testing and errors

- Limits to precision of inputs/outputs
  - 0.2 = binary 0.0011 recurring
  - PI is irrational
  - Floating point distribution is discrete input with spacing relative to log(x), $x_n - x_{n-1}$ = unit of least precision
  - Voltage sampled at particular bit accuracy
- Significance for testing
  - Inputs are rounded leading to inherent error in input
  - Outputs are rounded leading to inherent error in

Floating point distribution

- Floating point
  - Mantissa x 2^ exponent
  - For single precision
    - ULP = $2^{-24} \times 2^{EXPONENT}$
  - For double precision
    - ULP = $2^{-53} \times 2^{EXPONENT}$
- Largest ULP is
  - $2^{-24} \times 2^{127} = 2^{103}\ (single)$
  - $2^{-53} \times 2^{1023} = 2^{970}\ (double)$

Conditioning number

- Relative change/error in output = Conditioning number x relative change in error/input
  - $\frac{f(x+error)-f(x)}{f(x)} = C\,.\,\frac{error}{x}$
  - $f(x + error) - f(x) = C.\frac{error.f(x)}{x}$ (tolerance)
  - This means
    - C= $\frac{f(x+error)-f(x)}{f(x)}\,.\left(\frac{x}{error}\right)$
  - Example f(x)=x/(1-x)
    - x = 0.998 error = 0.0001, C=526
    - x = 0.999 error = 0.0001, C=1111

Conditioning number for differentiable functions and small error

- $Cn(x0) = x0 \,.\, \frac{F'(x0)}{F(x0)}$
- So for F(x)=x/1-x
- $F'(x) = \frac{1}{(1-x)^2}$ $\quad \frac{\mathrm{F}'(\mathrm{x})}{\mathrm{F}(\mathrm{x})} = \frac{1}{\mathrm{x}-1}$
- So
- Cn(x) = $\frac{x}{1-x}$

Condition number examples

- What is Cn for sin(x)
- F(x) = sin(x) F'(x) = cos(x)
- Cn = $x.\frac{cos(x)}{\sin(x)} = x.\frac{1}{\tan(x)}$
- What range is x ill-conditioned for
  - $\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \ \ldots > x$
  - None
- cos(x) F(x) = cos(x) F'(x) = -sin(x)
- Cn = $x.\frac{-\sin(x)}{\cos(\mathrm{x})} = -x.\tan(x)$
- So ill-conditioned when x= $\frac{\pi}{2}\ or\ \frac{3\pi}{2}$

Conditioning implications

- Example
  - When solving linear equations, the system is called ill-conditioned if a small change in the input causes a large change in solution
- So ill-conditioned designs/algorithms can cause
  - Instability of output
  - Problems with the behaviour of physical systems, imagine electronic circuit, each component can have a range of values due to component tolerances

Conditioning number and testing

- If 1 of your calculation steps has high condition number, this can lead to large errors on result
- Possible to have well conditioned function but with poorly conditioned step in algorithm

Testing numeric functions

- | Fcalc(x)- Ftrue(x) | < tolerance

- Fcalc(x) is the value produce by our program
- To calculate Ftrue we can use a arbitary precision maths package such as GMP
  - gmplib.org
- But what about the tolerance

Tolerance

- Function of Unit of Least Precision
  - This is dependent on X
- Function of the condition number
  - The greater the condition factor, the greater the effect of loss of precision
- In general
  - Tolerance <= Cn x ULP X f(x)/x
  - This is from slide 11

Types of function tests

- Golden value
  - $| Fcalc(x) - Ftrue(x) | < tolerance$
- Special value tests
  - $\sin(\pi) = 0 \;\; \cos(0) = 1$
- Identity tests
  - $\sin(x)^2 + \cos(x)^2 = 1$
  - Good but could hide relative errors, example identity above can hold if sin too large and cos too small
  - Useful because can automatically generate

Inverse function tests

- X = $\left(\sqrt{X}\right)^2$
- Asserts functions are consistent
- Note that
  - Some functions have ill-conditioned inversions (same problem applies with identity tests)

Limitations of identities

- $\sin(x)^2 + \cos(x)^2 = 1$
  - Doesn't test for accuracy at all so..
  - What if sin(x)==1 for every x and cos(x)==0 for every x
- X = $\left(\sqrt{X}\right)^2$
  - Better since only 1 function under test (assuming you already have a working multiply)

Intermediary results

Many functions produce large intermediary results

Example binomials for large N

Chance of throwing 450 heads out of 1000 throws of a coin = $\left(\frac{1000!}{450!550!}\right)\left(\frac{1}{2}\right)^{1000}$

Depending on the design of the algorithm this can lead to smaller or greater errors in the output, try computing 1000! With your calculator

Summary

- Numeric function testing relies on
  - The properties of the function being tested (condition number, argument type)
  - Having proper values for
    - Correct function output
    - Tolerance for the test

Summary

Functions can

- Special values   golden values
- Linked together to give identities
- Be tested using a previously tested inverse function