CS 450: Numerical Anlaysis

Chapter 1 – Scientific Computing
Lecture 2
Floating Point

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

January 25, 2018

Floating Point Numbers

- **Scientific Notation**

  Floating-point numbers are a computational realization of scientific notation,

  $$4.12165 \times 10^6, 2.145 \times 10^{-3}$$

  - *Scientific-notation provides a unique representation of any real number for a given amount of 'precision' (number of significant digits)*
  - *Normalized floating-point numbers are just a binary form of scientific notation,*

    $$1.01001 \times 2^5, 1.0110 \times 2^{-3}$$

- **Significand (Mantissa) and Exponent** Given x with s leading bits $x_0, \dots, x_{s-1}$

  $$fl(x) = \sum_{i=0}^{s-1} x_i 2^{k-i} = \underbrace{x_0.x_1 \dots x_{s-1}}_{\text{significand/mantissa}} \times 2^{\overbrace{\text{exponent}}^{k}}$$

  A floating point number's binary representation has $s-1$ significand bits (excluding $x_0 = 1$), some bits to represent the exponent, and a sign bit

Rounding Error

- **Maximum Relative Representation Error (Machine Epsilon)**
  - *If we have s significant digits in scientific notation, our error is bounded to variations of 1 in least significant digit, whose magnitude relative to the number we are trying to represent is 10^{1-s} in decimal and 2^{1-s} in binary*
  - *Formally, with s significant binary digits the relative representation error of positive real number x is (with $k = \lfloor \log_2(|x|) \rfloor$ and each $x_i \in \{0, 1\}$)*

$$x = \sum_{i=0}^{\infty} x_i 2^{k-i} = x_{\mathit{rem}} + \sum_{i=0}^{s-1} x_i 2^{k-i}, \quad \textit{where} \quad |x_{\mathit{rem}}/x| \le 2^{1-s}$$

  - *The maximum such error, 2^{1-s}, is called machine epsilon,*

$$\epsilon = \operatorname*{argmin}_{\epsilon > 0}(\mathit{fl}(1+\epsilon) = 1+\epsilon)$$

Rounding Error in Operations (I)

- **Addition and Subtraction**
  - *Subtraction is just negation of a sign bit followed by addition*
  - *Catastrophic cancellation occurs when the magnitude of the result is much smaller than the magnitude of both operands*
  - *Cancellation corresponds to losing significant digits, e.g.*

$$3.1423 \times 10^5 - 3.1403 \times 10^5 = 2.0 \times 10^2$$

  - *Generally, we can bound the error incurred during addition of two real numbers* x, y *in floating point (ignoring final rounding, which has relative error* ϵ*) as*

$$\frac{|(x+y) - (fl(x) + fl(y)|}{|x+y|} \leq \frac{\epsilon(|x| + |y|)}{|x+y|}$$

    by this we can also observe that the condition number of addition of x, y *i.e.* $f(x, y) = x + y$*, is* $\kappa_\infty(f) = (|x| + |y|)/|x+y|$
  - *Consequently, when* $x + y = 0$ *and* $x, y \neq 0$ *addition is ill-posed*

Rounding Error in Operations (II)

- **Multiplication and Division**
  - *Multiplication is a lot safer than addition in floating point*
  - *To analyze its error, we use a 2-term Taylor series approximation typical in relative error analysis*

$$f(\epsilon) = (1+n\epsilon)^k \approx f(0) + \frac{df}{d\epsilon}(0)\epsilon = 1 + kn\epsilon$$

  since ϵ is small, this linear approximation is accurate (to within $O(\epsilon^2)$)

  - *Aside from final rounding, we can bound the error in multiplication as*

$$\frac{|xy - fl(fl(x)fl(y))|}{|xy|} \leq \frac{|xy - (x(1+\epsilon)y(1+\epsilon))(1+\epsilon)|}{|xy|} \approx 3\epsilon$$

  - *Consequently, multiplication $f(x, y) = xy$ is always well-conditioned, $\kappa(f) \approx 3$*
  - *Division is multiplication by the reciprocal, and reciprocation is also well-conditioned*

Exceptional and Subnormal Numbers

- **Exceptional Numbers**

  We had mentioned that the leading bit in normalized floating point numbers is assumed to be 1, but how do represent 0?

  - *Exceptional floating point numbers are $0, -0, \infty, -\infty$, and NaN $= 0/0 = \infty - \infty$*

- **Subnormal (Denormal) Number Range**

  - *The range of magnitudes of normalized floating point numbers with an exponent range $[-e, e]$ is $[2^{-e}, 2^{e+1}(1-\epsilon/2)]$*
  - *For numbers of magnitude $< 2^{-e}$, the relative representation error is unbounded*
  - *Subnormal numbers are evenly spaced in $[-2^{-e}, 2^{-e}]$ with gaps of $\epsilon 2^{-e}$*
  - *Consequently, the absolute representation error in $[-2^{-e}, 2^{-e}]$ is at most $\epsilon 2^{-e}$*

- **Gradual Underflow: Avoiding underflow in addition**

  The main benefit of subnormal numbers is that for any machine numbers (floating-point numbers) x and y, $fl(x-y) = 0$ if and only if $x = y$, since the gap between any two representable numbers is $|x-y| \geq \epsilon 2^{-e}$