CSC 338 Lecture 2

Floating-Point Numbers

Advantages: - FP numbers is a way to store continuous quantities using discrete hardware
- Each FP number takes same amount of storage

Disadvantages: - FP arithmetic is inexact
↳ overflow / underflow
↳ loss of accuracy
↳ cancellation / catastrophic cancellation.

Our treatment of FP numbers will be more general than in CSC 207

Intuition: floating-point numbers are like scientific notation.

S.A. of earth. 510 072 000 km^2.

scientific notation: 5.10072×10^8 km^2.

fewer precision: $\underbrace{5.101}_{\#\text{digits} = \text{precision}} \times \underbrace{10}_{\text{base}}{}^{8}$ km^2 ← in base 10

one digit before decimal

exponent denote position of decimal

Def: A Floating-Point Number System $F(\beta, p, L, U)$ is characterized by integers
β — base
p — precision
$[L, U]$ — exponent range.

In this system, a real number x is represented as:

$$x \approx \pm \left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \dots + \frac{d_{p-1}}{\beta^{p-1}}\right) \beta^E.$$

where $d_i \in \{0, 1, \dots, \beta - 1\}$

$E \in \{L, L+1, \dots, U-1, U\}$

The digits $d_0 d_1 \ldots d_{p-1}$ is called the mantissa. ②.

$d_1 \ldots d_{p-1}$ is called the fraction.

E is called the exponent

eg// In the system $F(\beta=10,\ p=3,\ L=-100,\ U=100)$ we can represent the S.A. of earth as.

$$+\left(\underset{d_0}{5}+\underset{d_1}{\frac{1}{10}}+\underset{d_2}{\frac{0}{10^2}}\right)\cdot 10^{\overset{E}{8}} \iff \underbrace{5.10}_{\text{base 10 \#s.}} \times \underline{10}^8$$

Q what about 0.51×10^9 or $+\left(0+\frac{5}{10}+\frac{1}{10^2}\right)\cdot 10^9$?

Def. A F.P. system is normalized if we enforce the additional rule that the leading digit $d_0 \neq 0$ unless the number represented is 0.

$\Rightarrow$ There is a unique normalized repr for each nonzero value.

$\Rightarrow$ If $\beta=2$, then $d_0=1$ for all nonzero values, so we don't need to store d_0.

eg//. IEEE SP System $F(\beta=2,\ p=24,\ L=-126,\ U=127)$. What are the smallest and largest positive number that we can represent in this system? normalized F.P.

Smallest: set $d_0=1$, $d_i=0$ for $i\neq 0$, $E=-126$.

$$\Rightarrow +\left(1+\frac{0}{2}+\ldots+\frac{0}{2^{23}}\right)\times 2^{-126} = 2^{-126}.$$

largest: Set $d_i=1$, $E=127$

$$\Rightarrow +\left(1+\frac{1}{2}+\ldots+\frac{1}{2^{23}}\right)\times 2^{127}$$

Def The underflow level (UFL) is the smallest positive normalized FP number in $F(\beta, p, L, U)$. Can show $UFL=\beta^L$

The overflow level (OFL) is the largest positive normalized FP number in $F(\beta, p, L, U)$. Can show $OFL = (1-\beta^{-p})\beta^{U+1}$

eg// How many FP numbers are there in a normalized FP system $F(\beta, p, L, U)$?

$$\underbrace{2}_{\text{sign}} \times \underbrace{(\beta - 1)}_{\text{leading digit}} \underbrace{\beta^{p-1}}_{\text{rest of mantissa}} \underbrace{(U - L + 1)}_{\text{exponent}} + \underbrace{1}_{\text{zero}}.$$

Def: Numbers that are exactly representable in a ~~normalized~~ F.P. system are called machine numbers.

When we plot machine numbers on the number line, there is a "gap" around zero due to normalization.

Def. In a subnormal or denormalized FP system, we relax normalization and allow leading zeros ($d_0 = 0$) when $E = L$. These new FP numbers are called subnormal or denormalized. This augmented system exhibit gradual underflow.

(Break: 4:05)

Rounding

We represent $x \in \mathbb{R}$ in a F.P. system by approximating x with a "nearby" machine number $fl(x)$ via rounding.

Two method of rounding.

1. Chopping "Round toward zero" ~ truncate the base β expansion of x after $(p-1)$ digits
2. Round to nearest. Choose $fl(x)$ to minimize $|fl(x) - x|$

eg// want to represent $x = 5.10072 \times 10^8$ in $F(\beta=10, p=4, L=-100, U=100)$

Chop — 5.100×10^8

Nearest — 5.101×10^8

The machine precision ϵ_{mach} characterizes the accuracy of a F.P. system. Can be defined in a few ways.

Def$_1$ ϵ_{mach} is the maximum relative error in representing a nonzero $x \in \mathbb{R}$ with $fl(x)$

Def$_2$ ϵ_{mach} is the smallest ϵ with $fl(1+\epsilon) > 1$

Q: what is the machine precision for $F(\beta, p, L, U)$?

A: If we round by chopping. $\epsilon_{mach} = \beta^{1-p} = (\beta^{-p+1})$

to nearest. $\epsilon_{mach} = \frac{1}{2}\beta^{1-p}$.

Floating-Point Arithmetics

Floating-Point Arithmetic is inexact.

multiplication. Product of two FP numbers with precision p will have $2p$ digits $\Rightarrow$ need to round.

eg//. In $F(\beta=10, p=2, L=-10, U=10)$

$x = 5.9 \times 10^2$

$y = 6.1 \times 10^1$

$$\begin{array}{rl} 5.9 & \times 10^2 \\ 6.1 & \times 10^1 \\ \hline 59 & \\ 354 & \\ \hline \underbrace{3.599}_{2p = 4 \text{ digits.}} & \times 10^4 \end{array} \Rightarrow 3.6 \times 10^4.$$

Divisions Also inexact. Long division.

Quotient can have many more digits, potentially infinite.

eg// $1 \div 3$ has infinite # digits in $\beta = 10$.

Addition (subtraction).

shift the mantissa to match the exponent, then add/subtract along the column.

eg// In $F(\beta=10, p=4, L=-10, U=10)$

$x = 1.924 \times 10^2$

$y = 6.357 \times 10^{-1}$

$x+y$:

$$\begin{array}{rl} 1.924 & \times 10^2 \\ 6.357 & \times 10^{-1} \\ \hline \boxed{1.930}\underline{357}. & \times 10^2 \end{array}$$

lose this info due to rounding.

$x+y \Rightarrow 1.930 \times 10^2$

eg// In $F(\beta = 5, p=3, L=-10, U=10)$

base 5

$x = 2.44 \times 5^0$

$y = 3.33 \times 5^1$

$$\begin{array}{rl} .2.44 & \times 5^0 \\ 3.33 & \times 5^1 \\ \hline 4.12.4 & \times 5^1 \end{array}$$

Round to nearest

4.13×5^1

Ideally a floating-point operation flop. (eg// F.P. addition)

vs. its arbitrary precision op (eg// real addition)

should have $\underline{fl(x) \text{ flop } fl(x) = fl(x \text{ op } y)}$

In most computers, this is true!

Relative error of each computation is bounded by ϵ_{mach}

But there are still issues to be aware of

eg// normal law of arithmetic don't hold.

$\Rightarrow$ Floating-Point Addition is not associative!

eg// Choose $\epsilon = 0.6\, \epsilon_{mach}$. So $1 +_{fl} \epsilon = 1$ (becaus $\epsilon < \epsilon_{mach}$)

So $(1 +_{fl} \epsilon) +_{fl} \epsilon = 1$

$1 +_{fl} (\epsilon +_{fl} \epsilon) > 1$

eg// Choose $\epsilon = 0.6\, \epsilon_{mach}$.

$$(1 +_{fl} \epsilon) -_{fl} (1 -_{fl} \epsilon) = 1 -_{fl} 1 = 0.$$

~~eg~~.

FP arithmetic can cause an overflow when the result has $E > U$, an underflow when the result has $E < L$

eg// $\sum_{i=1}^{\infty} \frac{1}{n}$. should diverge, but converges in FP arithmetic.

Potential Reason : ~~overflow~~ because Σ gets too large.

~~underflow~~ because $\frac{1}{n}$ gets too small

✓ because $\frac{1}{n}$ becomes insignificant compare to the Σ so far

Cancellation.

Subtracting two p-digit numbers with similar sign and magnitude. yield result with much fewer than p digits

eg//. $F(\beta = 10,\ p = 6,\ L = -10,\ U = 10)$

$$\begin{array}{r} 1.92403 \times 10^{2} \\ -\ 1.92275 \times 10^{2} \\ \hline \underbrace{1.28}_{}000 \quad \times 10^{-1} \end{array}$$

only 3 significant digits (started w/ 6)

eg// Computing standard deviation

$$\sum_{i=1}^{N} (\underset{\text{data point}}{x_i} - \underset{\text{mean}}{\bar{x}})^2 = \underbrace{\underbrace{\sum_{i=1}^{N} x_i^2}_{\text{large}} - \underbrace{N\bar{x}^2}_{\text{large}}}_{\text{difference much smaller}}$$

Catastrophic Cancellation

In some cases, cancellation can be so bad that the solution has <u>no</u> correct significant digits!

eg// $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$

For x that is large and negative, computing e^x using its Taylor Series give disasterous results.

For $x=40$, e^{-40} is small but with

$$e^{-40} = 1 + (-40) + \frac{(-40)^2}{2!} + \frac{(-40)^3}{3!} + \dots$$

each term is large, and adjacent terms have opposite signs $\Rightarrow$ cancellation.

moral of today's lecture:

- FP usually good enough
- avoid subtracting two large values to get a small result.