AMS 147 Computational Methods and Applications

Lecture 06
Copyright by Hongyun Wang, UCSC

Recap of Lecture 5:

Newton's method for solving non-linear systems $\vec{f}(\vec{x}) = 0$

$\vec{x}_{n+1} = \vec{x}_n + \Delta\vec{x}_n$ where $\Delta\vec{x}_n$ is the solution of $\nabla\vec{f}(\vec{x}_n)\Delta\vec{x}_n = -\vec{f}(\vec{x}_n)$

Floating point representation

In computers, a <u>non-zero</u> real number x is represented as

$$\mathrm{fl}(x) = \sigma \times (.a_1 a_2 \cdots a_t)_\beta \times \beta^p$$

Mathematical meaning:

$$\sigma \times (.a_1 a_2 \cdots a_t)_\beta \times \beta^p = \sigma \times \left(\frac{a_1}{\beta} + \frac{a_2}{\beta^2} + \cdots + \frac{a_t}{\beta^t}\right)_\beta \times \beta^p$$

Machine precision: $\beta^{-(t-1)}$

The smallest number above 1 that can be represented exactly is

$$\mathrm{fl}\left(1 + \beta^{-(t-1)}\right) = 1 + \beta^{-(t-1)}$$

For $1 < x < 1 + \beta^{-(t-1)}$,

$$\mathrm{fl}(x) \neq x$$

The middle point between 1 and $1 + \beta^{-(t-1)}$ is $\boxed{1 + \beta^{-t}}$.

$1 \le x < 1 + \beta^{-t}$ ==> $\mathrm{fl}(x) = 1$

$x > 1 + \beta^{-t}$ ==> $\mathrm{fl}(x) > 1$

(Draw the real axis to show 1, $1 + \beta^{-(t-1)}$ **and the middle point).**

The largest number below 1 that can be represented exactly is

$$\mathrm{fl}\left(1 - \beta^{-t}\right) = 1 - \beta^{-t}$$

For $1 + \beta^{-t} < x < 1$,

$$\mathrm{fl}(x) \neq x$$

The middle point between $1 - \beta^{-t}$ and 1 is $\boxed{1 - \beta^{-(t+1)}}$.

$1-\beta^{-(t+1)} < x \le 1 \quad ==> \quad \mathrm{fl}(x) = 1$

$x < 1-\beta^{-(t+1)} \quad ==> \quad \mathrm{fl}(x) < 1$

(Draw the real axis to show $1-\beta^{-t}$, 1 and the middle point).

Example: β = 2, *t* = 53

Find whether or not $\mathrm{fl}\left(1-2^{-50}\right) = 1$.

The middle point between $1-\beta^{-t}$ and 1 is

$$1-\beta^{-(t+1)} = 1-2^{-54}$$

We compare $1-2^{-50}$ with the middle point.

$$1-2^{-50} < 1-2^{-54}$$

$==> \quad \mathrm{fl}\left(1-2^{-50}\right) < 1$

For $1-2^{-60}$, we have

$$1-2^{-54} < 1-2^{-60} < 1$$

$==> \quad \mathrm{fl}\left(1-2^{-60}\right) = 1$

Round-off error

Round-off error is the difference between fl(*x*) and *x*.

Case 1: Suppose we do truncating.

If we are allowed to use infinitely many bits in the mantissa, *x* can be represented exactly as

$$x = \sigma \times \left(.a_1 a_2 \cdots a_t\, a_{t+1} \cdots\right)_\beta \times \beta^p$$

The floating point representation obtained by truncating is

$$\mathrm{fl}(x) = \sigma \times \left(.a_1 a_2 \cdots a_t\right)_\beta \times \beta^p$$

$$==> \quad \mathrm{fl}(x) - x = -\sigma \times \left(.\underbrace{0 \cdots 0}_{t}\, a_{t+1}\, a_{t+2} \cdots\right)_\beta \times \beta^p$$

$$= -\sigma \times \left(.a_{t+1}\, a_{t+2} \cdots\right)_\beta \times \beta^{p-t}$$

The absolute error (if we do truncating) is

$$\left|\, \mathrm{fl}(x) - x \,\right| = \left(.a_{t+1}\, a_{t+2} \cdots\right)_\beta \times \beta^{p-t} \le \beta^{p-t}$$

Here we have used $(.a_{t+1}\,a_{t+2}\cdots)_\beta \le 1$.

The relative error (if we do truncating) is

$$\frac{|\,\mathrm{fl}(x)-x\,|}{|\,x\,|} \le \frac{\beta^{p-t}}{(.a_1\,a_2\cdots a_t\,a_{t+1}\cdots)_\beta \times \beta^p} \le \frac{\beta^{p-t}}{\beta^{-1}\cdot\beta^p} = \beta^{-(t-1)}$$

Here we have used $(.a_1\,a_2\cdots a_t\,a_{t+1}\cdots)_\beta \ge (.1)_\beta = \beta^{-1}$

<u>Summary of case #1:</u>

Suppose we do <u>truncating</u>. We have

$$|\,\mathrm{fl}(x)-x\,| \le \beta^{p-t}$$

$$\frac{|\,\mathrm{fl}(x)-x\,|}{|\,x\,|} \le \beta^{-(t-1)}$$

<u>Case 2:</u> Suppose we do rounding. We have

$$|\,\mathrm{fl}(x)-x\,| \le \frac{1}{2}\beta^{p-t}$$

$$\frac{|\,\mathrm{fl}(x)-x\,|}{|\,x\,|} \le \frac{1}{2}\beta^{-(t-1)}$$

That is, the bound of $|\,\mathrm{fl}(x)-x\,|$ is halved when we switch from truncating to rounding.

This can be illustrated by looking at how real numbers between 1 and $1+\beta^{-(t-1)}$ are stored in the floating-point representation system)

(Draw the real axis with 1 and $1+\beta^{-(t-1)}$**)**

<u>A mathematical form of fl(*x*)</u> for error analysis

We can write $\mathrm{fl}(x)$ as

$$\mathrm{fl}(x) = x + \mathrm{fl}(x) - x = x + x\cdot\frac{\mathrm{fl}(x)-x}{x} = x\left(1+\frac{\mathrm{fl}(x)-x}{x}\right)$$

Let

$$\varepsilon = \frac{\mathrm{fl}(x)-x}{x}.$$

We have

$$|\varepsilon| = \left|\frac{\mathrm{fl}(x) - x}{x}\right| \le \frac{1}{2}\beta^{-(t-1)}.$$

We write $\mathrm{fl}(x)$ as

$$\mathrm{fl}(x) = x\left(1 + \frac{\mathrm{fl}(x) - x}{x}\right) = x(1+\varepsilon)$$

Thus, we have

$$\boxed{\mathrm{fl}(x) = x(1+\varepsilon), \qquad |\varepsilon| \le \frac{1}{2}\beta^{-(t-1)}}$$

Note: This form of $\mathrm{fl}(x)$ is very useful in error analysis.

IEEE double precision floating point representation

$$\mathrm{fl}(x) = \sigma \times (.a_1 a_2 \cdots a_t)_\beta \times \beta^p$$

$$\beta = 2, \qquad t = 53$$

$$(p + bias) = (b_k b_{k-1} \cdots b_1)_\beta,$$

$$bias = 1023, \qquad k = 11$$

$$L \le p \le U$$

$$L = -1022, \qquad U = 1023$$

A few items about IEEE double precision:

- $\mathrm{fl}(x)$ occupies

  1 + (t – 1) + k = 64 bits = 8 bytes (1 byte = 8 bits).

- Machine precision:

  $$\beta^{-(t-1)} = 2^{-52} \approx 2.22 \times 10^{-16}$$

- Round-off error:

  $$\mathrm{fl}(x) = x(1+\varepsilon)$$

  $$|\varepsilon| \le \frac{1}{2}\beta^{-(t-1)} = 2^{-53} \approx 1.11 \times 10^{-16}$$

- Question: How is “0” represented?

  The range of p is

  $$-1022 \le p \le 1023$$

  $$bias = 1023$$

==> $1 \le (p + bias) \le 2046$

p is stored as

$$(p + bias) = (b_{11}\, b_{10} \cdots b_1)_\beta$$

The smallest of $(b_{11}\, b_{10} \cdots b_1)_\beta$ is

$$\Big(\underbrace{0\,0 \cdots 0}_{11}\Big)_\beta = 0$$

The largest of $(b_{11}\, b_{10} \cdots b_1)_\beta$ is

$$\Big(\underbrace{1\,1 \cdots 1}_{11}\Big)_\beta = 1 + 2 + 2^2 + 2^{10} = 2^{11} - 1 = 2047$$

==> $0 \le (b_{11}\, b_{10} \cdots b_1)_\beta \le 2047$

We compare the range of $(p + bias)$ and the range of $(b_{11}\, b_{10} \cdots b_1)_\beta$

$$1 \le p + bias \le 2046$$

$$0 \le (b_{11}\, b_{10} \cdots b_1)_\beta \le 2047$$

We see that $(b_{11}\, b_{10} \cdots b_1)_\beta = (0\,0 \cdots 0)_\beta$ and $(b_{11}\, b_{10} \cdots b_1)_\beta = (1\,1 \cdots 1)_\beta$ are not used in storing p.

They are used to store special numbers.

$(b_{11}\, b_{10} \cdots b_1)_\beta = (0\,0 \cdots 0)_\beta$ is used to store "0" (the real number zero).

$(b_{11}\, b_{10} \cdots b_1)_\beta = (1\,1 \cdots 1)_\beta$ is used to store arithmetic exceptions (*Inf*, –*Inf*, *NaN*)

Overflow and underflow

In the IEEE double precision representation,

$$\mathrm{fl}(x) = \sigma \times (.a_1\, a_2 \cdots a_t)_\beta \times \beta^p$$

$$L \le p \le U$$

The largest number (in absolute value) is

$$B = (.1\,1 \cdots 1)_\beta \cdot \beta^U \approx \beta^U = 2^{1023} \approx 10^{308}$$

The smallest non-zero number (in absolute value) is

$$b = (.1\,0\cdots 0)_\beta \cdot \beta^L = \beta^{L-1} = 2^{-1022-1} \approx 10^{-308}$$

Overflow:

If $|x| > B$, then $\mathrm{fl}(x) = \inf$.

This is called overflow.

Note: overflow is a fatal error.

Underflow:

If $|x| < \frac{b}{2}$, then $\mathrm{fl}(x) = 0$.

This is called underflow.

Note: underflow is a non-fatal error.

Now let us go through two simple examples to see the difference between the exact arithmetic and finite precision arithmetic.

Example:

Exact arithmetic:

$$1 + 2^{-54}$$

IEEE Double precision representation:

$$\mathrm{fl}(1 + 2^{-54}) = 1$$

We can see $\mathrm{fl}(1 + 2^{-54}) = 1$ by drawing the real axis.

In IEEE double precision representation, the smallest number above 1 is

$1 + \beta^{-(t-1)} = (1 + 2^{-52})$.

The middle point between 1 and $1 + \beta^{-(t-1)}$ is $1 + \beta^{-t} = (1 + 2^{-53})$.

$$1 < 1 + 2^{-54} < 1 + 2^{-53}$$

==> $\mathrm{fl}(1 + 2^{-54}) = 1$

Note: This example demonstrates the difference between the exact arithmetic and a finite precision arithmetic. A finite precision arithmetic has round-off errors while the exact arithmetic does not. As we will see below, if we are not careful, the effect of round-off errors can be devastating.

Example:

Exact arithmetic:

$$\frac{\left(1+2^{-54}\right)-1}{2^{-54}}=1$$

IEEE Double precision FPR:

$$\frac{\text{fl}\left(1+2^{-54}\right)-\text{fl}\left(1\right)}{\text{fl}\left(2^{-54}\right)}=\frac{1-1}{2^{-54}}=0$$

Note: In this example, the result of IEEE Double precision FPR is 100% different from the result of the exact arithmetic.

Let us see two more examples of determining whether or not $\text{fl}(x)=1$.

Example: Let $a = 2^{-30}$.

Find whether or not $\text{fl}\left(\cos\left(a\right)\right)=1$ in IEEE double precision representation.

Taylor expansion of $\cos(a)$:

$$\cos\left(a\right)=1-\frac{1}{2}a^2+O\left(a^4\right)$$

$$\approx 1-\frac{1}{2}a^2=1-2^{-61}<1$$

The middle point between $1-\beta^{-t}$ and 1 is

$$1-\beta^{-(t+1)}=1-2^{-54}$$

We compare $1-2^{-61}$ with the middle point.

$$1-2^{-54}<1-2^{-61}<1$$

$$\Longrightarrow \quad \text{fl}\left(\cos\left(a\right)\right)=\text{fl}\left(1-2^{-61}\right)=1$$

Example: Let $b = 2^{-50}$.

Find whether or not $\text{fl}\left(\exp\left(b\right)\right)=1$ in IEEE double precision representation.

Taylor expansion of $\exp(b)$:

$$\exp\left(b\right)=1+b+O\left(b^2\right)$$

$$\approx 1+b=1+2^{-50}>1$$

The middle point between $1+\beta^{-(t-1)}$ and 1 is

$$1+\beta^{-t}=1+2^{-53}$$

We compare $1+2^{-50}$ with the middle point.

$$1 + 2^{-50} > 1 + 2^{-53}$$

==> $\mathrm{fl}\left(\exp(b)\right) = \mathrm{fl}\left(1 + 2^{-50}\right) > 1$

(**Go through sample codes in assignment #2**)