

Mathematical preliminaries and error analysis

Tsung-Ming Huang

Department of Mathematics
National Taiwan Normal University, Taiwan

September 12, 2015



Outline

- 1 Round-off errors and computer arithmetic**
 - IEEE standard floating-point format
 - Absolute and Relative Errors
 - Machine Epsilon
 - Loss of Significance

- 2 Algorithms and Convergence**
 - Algorithm
 - Stability
 - Rate of convergence



Example 4

What is the decimal number of the machine number

0100000010011111111111111111111111111111?

- 1 The final 23 bits specify that the mantissa is

$$\begin{aligned} f &= 0 \cdot (2)^{-1} + 0 \cdot (2)^{-2} + 1 \cdot (2)^{-3} + \cdots + 1 \cdot (2)^{-23} \\ &= 0.2499998807907105. \end{aligned}$$

- 2 Consequently, this machine number precisely represents the decimal number

$$\begin{aligned} (-1)^s 2^{c-127} (1 + f) &= 2^2 \cdot (1 + 0.2499998807907105) \\ &= 4.999999523162842. \end{aligned}$$



Example 5

What is the decimal number of the machine number

01000000101000000000000000000001?

- 1 The final 23 bits specify that the mantissa is

$$\begin{aligned} f &= 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + \dots + 0 \cdot 2^{-22} + 1 \cdot 2^{-23} \\ &= 0.2500001192092896. \end{aligned}$$

- 2 Consequently, this machine number precisely represents the decimal number

$$\begin{aligned} (-1)^s 2^{c-127} (1 + f) &= 2^2 \cdot (1 + 0.2500001192092896) \\ &= 5.000000476837158. \end{aligned}$$



Summary

Above three examples

0100000010011111111111111111111111111111111 \Rightarrow 4.999999523162842

010000001010000000000000000000000000000000 \Rightarrow 5

010000001010000000000000000000000000000001 \Rightarrow 5.000000476837158

- Only a relatively **small subset** of the real number system is used for the representation of all the real numbers.
- This subset, which are called the *floating-point numbers*, contains only rational numbers, both positive and negative.
- When a number can not be represented exactly with the fixed finite number of digits in a computer, a **near-by** floating-point number is chosen for approximate representation.



The smallest positive number

Let $s = 0$, $c = 1$ and $f = 0$ which is equivalent to

$$2^{-126} \cdot (1 + 0) \approx 1.175 \times 10^{-38}$$

The largest number

Let $s = 0$, $c = 254$ and $f = 1 - 2^{-23}$ which is equivalent to

$$2^{127} \cdot (2 - 2^{-23}) \approx 3.403 \times 10^{38}$$

Definition 6

If a number x with $|x| < 2^{-126} \cdot (1 + 0)$, then we say that an *underflow* has occurred and is generally set to zero.

If $|x| > 2^{127} \cdot (2 - 2^{-23})$, then we say that an *overflow* has occurred.

Format of floating-point number

$$(-1)^s \times (1 + f) \times 2^{c-1023}$$

The smallest positive number

Let $s = 0$, $c = 1$ and $f = 0$ which is equivalent to

$$2^{-1022} \cdot (1 + 0) \approx 2.225 \times 10^{-308}.$$

The largest number

Let $s = 0$, $c = 2046$ and $f = 1 - 2^{-52}$ which is equivalent to

$$2^{1023} \cdot (2 - 2^{-52}) \approx 1.798 \times 10^{308}.$$



Chopping and rounding

For any real number x , let

$$x = \pm 1.a_1 a_2 \cdots a_t a_{t+1} a_{t+2} \cdots \times 2^m,$$

denote the normalized scientific binary representation of x .

- 1 **chopping**: simply discard the excess bits a_{t+1}, a_{t+2}, \dots to obtain

$$fl(x) = \pm 1.a_1 a_2 \cdots a_t \times 2^m.$$

- 2 **rounding**: add $2^{-(t+1)} \times 2^m$ to x and then chop the excess bits to obtain a number of the form

$$fl(x) = \pm 1.\delta_1 \delta_2 \cdots \delta_t \times 2^m.$$

In this method, if $a_{t+1} = 1$, we add 1 to a_t to obtain $fl(x)$, and if $a_{t+1} = 0$, we merely chop off all but the first t digits.



Definition 7 (Roundoff error)

The error results from replacing a number with its floating-point form is called *roundoff error* or *rounding error*.

Definition 8 (Absolute Error and Relative Error)

If x is an approximation to the exact value x^* , the **absolute error** is $|x^* - x|$ and the **relative error** is $\frac{|x^* - x|}{|x^*|}$, provided that $x^* \neq 0$.

Example 9

(a) If $x^* = 0.3000 \times 10^{-3}$ and $x = 0.3100 \times 10^{-3}$, then the absolute error is 0.1×10^{-4} and the relative error is 0.3333×10^{-1} .

(b) If $x^* = 0.3000 \times 10^4$ and $x = 0.3100 \times 10^4$, then the absolute error is 0.1×10^3 and the relative error is 0.3333×10^{-1} .



Remark 1

As a measure of accuracy, the absolute error may be misleading and the relative error more meaningful.

Definition 10

The number x is said to approximate x^* to t **significant digits** if t is the largest nonnegative integer for which

$$\frac{|x - x^*|}{|x^*|} \leq 5 \times 10^{-t}.$$



Absolute and Relative Errors

- If the floating-point representation $fl(x)$ for the number x is obtained by using t digits and chopping procedure, then the relative error is

$$\begin{aligned} \frac{|x - fl(x)|}{|x|} &= \frac{|0.00 \cdots 0a_{t+1}a_{t+2} \cdots \times 2^m|}{|1.a_1a_2 \cdots a_t a_{t+1}a_{t+2} \cdots \times 2^m|} \\ &= \frac{|0.a_{t+1}a_{t+2} \cdots|}{|1.a_1a_2 \cdots a_t a_{t+1}a_{t+2} \cdots|} \times 2^{-t}. \end{aligned}$$

The minimal value of the denominator is 1. The numerator is bounded above by 1. As a consequence

$$\left| \frac{x - fl(x)}{x} \right| \leq 2^{-t}.$$



Absolute and Relative Errors

- If t -digit rounding arithmetic is used and
 - $a_{t+1} = 0$, then $fl(x) = \pm 1.a_1a_2 \cdots a_t \times 2^m$. A bound for the relative error is

$$\frac{|x - fl(x)|}{|x|} = \frac{|0.a_{t+1}a_{t+2} \cdots|}{|1.a_1a_2 \cdots a_t a_{t+1}a_{t+2} \cdots|} \times 2^{-t} \leq 2^{-(t+1)},$$

since the numerator is bounded above by $\frac{1}{2}$ due to $a_{t+1} = 0$.

- $a_{t+1} = 1$, then $fl(x) = \pm(1.a_1a_2 \cdots a_t + 2^{-t}) \times 2^m$. The upper bound for relative error becomes

$$\frac{|x - fl(x)|}{|x|} = \frac{|1 - 0.a_{t+1}a_{t+2} \cdots|}{|1.a_1a_2 \cdots a_t a_{t+1}a_{t+2} \cdots|} \times 2^{-t} \leq 2^{-(t+1)},$$

since the numerator is bounded by $\frac{1}{2}$ due to $a_{t+1} = 1$.

Therefore the relative error for rounding arithmetic is

$$\left| \frac{x - fl(x)}{x} \right| \leq 2^{-(t+1)} = \frac{1}{2} \times 2^{-t}.$$



Definition 11 (Machine epsilon)

The floating-point representation, $fl(x)$, of x can be expressed as

$$fl(x) = x(1 + \delta), \quad |\delta| \leq \varepsilon_M, \quad (1)$$

where $\varepsilon_M \equiv 2^{-t}$ is referred to as the *unit roundoff error* or *machine epsilon*.

Single precision IEEE standard floating-point format

The mantissa f corresponds to 23 binary digits (i.e., $t = 23$), the machine epsilon is

$$\varepsilon_M = 2^{-23} \approx 1.192 \times 10^{-7}.$$

This approximately corresponds to **7** accurate decimal digits



Double precision IEEE standard floating-point format

The mantissa f corresponds to 52 binary digits (i.e., $t = 52$), the machine epsilon is

$$\varepsilon_M = 2^{-52} \approx 2.220 \times 10^{-16}.$$

which provides between **15** and **16** decimal digits of accuracy.

Summary of IEEE standard floating-point format

	single precision	double precision
ε_M	1.192×10^{-7}	2.220×10^{-16}
smallest positive number	1.175×10^{-38}	2.225×10^{-308}
largest number	3.403×10^{38}	1.798×10^{308}
decimal precision	7	16



Machine Epsilon

- Let \odot stand for any one of the four basic arithmetic operators $+$, $-$, $*$, \div .
- Whenever two **machine numbers** x and y are to be combined arithmetically, the computer will produce $fl(x \odot y)$ instead of $x \odot y$.
- Under (1), the relative error of $fl(x \odot y)$ satisfies

$$fl(x \odot y) = (x \odot y)(1 + \delta), \quad \delta \leq \varepsilon_M, \quad (2)$$

where ε_M is the unit roundoff.

- But if x, y are **not** machine numbers, then they must first be rounded to floating-point format before the arithmetic operation and the resulting relative error becomes

$$fl(fl(x) \odot fl(y)) = (x(1 + \delta_1) \odot y(1 + \delta_2))(1 + \delta_3),$$

where $\delta_i \leq \varepsilon_M, i = 1, 2, 3$.



Loss of Significance

- One of the most common error-producing calculations involves the cancellation of significant digits due to the **subtraction of nearly equal numbers** or the **addition of one very large number and one very small number**.
- Sometimes, loss of significance can be avoided by rewriting the mathematical formula.

Example 12

The quadratic formulas for computing the roots of $ax^2 + bx + c = 0$, when $a \neq 0$, are

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Consider the quadratic equation $x^2 + 62.10x + 1 = 0$ and discuss the numerical results.



Solution

- Using the quadratic formula and 8-digit rounding arithmetic, one can obtain

$$x_1 = -0.01610723 \quad \text{and} \quad x_2 = -62.08390.$$

- Now we perform the calculations with 4-digit rounding arithmetic. First we have

$$\sqrt{b^2 - 4ac} = \sqrt{62.10^2 - 4.000} = \sqrt{3856 - 4.000} = 62.06,$$

and

$$fl(x_1) = \frac{-62.10 + 62.06}{2.000} = \frac{-0.04000}{2.000} = -0.02000.$$

The relative error in computing x_1 is

$$\frac{|fl(x_1) - x_1|}{|x_1|} = \frac{|-0.02000 + 0.01610723|}{|-0.01610723|} \approx 0.2417 \leq 5 \times 10^{-1}$$



Loss of Significance

- In calculating x_2 ,

$$fl(x_2) = \frac{-62.10 - 62.06}{2.000} = \frac{-124.2}{2.000} = -62.10,$$

and the relative error in computing x_2 is

$$\frac{|fl(x_2) - x_2|}{|x_2|} = \frac{|-62.10 + 62.08390|}{|-62.08390|} \approx 0.259 \times 10^{-3} \leq 5 \times 10^{-4}.$$

- In this equation, $b^2 = 62.10^2$ is much larger than $4ac = 4$. Hence b and $\sqrt{b^2 - 4ac}$ become two nearly equal numbers. The calculation of x_1 involves the subtraction of two nearly equal numbers.
- To obtain a more accurate 4-digit rounding approximation for x_1 , we change the formulation by rationalizing the numerator, that is,

$$x_1 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}.$$



Then

$$fl(x_1) = \frac{-2.000}{62.10 + 62.06} = \frac{-2.000}{124.2} = -0.01610.$$

The relative error in computing x_1 is now reduced to 6.2×10^{-4}



Example 13

Let

$$p(x) = x^3 - 3x^2 + 3x - 1,$$

$$q(x) = ((x - 3)x + 3)x - 1.$$

Compare the function values at $x = 2.19$ with using three-digit arithmetic.



