Round-off error analysis

Input error $x \to fl(x)$

$fl(x) = \pm 0.d_1 d_2 \cdots d_n \beta^b$ → Chopping

$x = \pm 0.d_1 d_2 \cdots d_n d_{n+1} \cdots \beta^b$ → rounding

Round-off $|x - fl(x)| \le \frac{1}{2}\beta^{b-n}$ $x \neq 0,\ d_1 \neq 0$

$x \neq 0.$ $\left|\frac{x - fl(x)}{x}\right| \le \frac{\frac{1}{2}\beta^{b-n}}{0.1 \cdot \beta^b} = \boxed{\frac{1}{2}\beta^{-n+1}} = \varepsilon$

ε is called the precision, epsilon, accuracy.

$fl(x)$ is the normalized floating number

$\beta = 10.$ $\varepsilon = \frac{1}{2}10^{-n+1}$ $n = 23$ ε

How to represent computers in terms of real number system.

Thm: (Wilkinson) If $x \in R$, $fl(x) = x(1+\delta_x) = x + x\delta$ is a perturbation of true value of x

$x - fl(x) = -x\delta$ $\quad |x - fl(x)| = |x||\delta|$

relative error $\frac{x - fl(x)}{x} = \frac{-x\delta}{x} = -\delta$ $\quad x = 3.14$

$|\delta| \le \frac{1}{2}\beta^{-n+1} = \varepsilon$

$\frac{|\pi - fl(\pi)|}{\pi} \le \varepsilon = \frac{1}{2}\beta^{-n+1}$

Conclusion: When we input a number into a computer system, the relative error is bounded by the precision

Operation error analysis $\pm$ $\times$ / logic oper.

Let x and y are numbers in a computer system. and, if, or, no errors

$$fl(x \circ y) = (x \circ y)(1+\delta_1) \quad \text{Theorem.}$$

$|\delta_1| \le \varepsilon$. $\circ$, +, −, $\times$ $\div$, $\sqrt{\ }$

$x = 0.d_1 d_2 \cdots d_n \cdot \beta^{b_1}$

$y = 0.\bar{d}_1 \bar{d}_2 \cdots \bar{d}_n \beta^{b_2}$ $b_1 \neq b_2$, $d_1 \neq 0$

$0.d_1 d_2 \cdots d_n 10^{+2}$ $\bar{d}_1 \neq 0$

$0.\bar{d}_1 \bar{d}_2 \cdots \bar{d}_n \cdot 10^0$

x, y are not in the computer system, $0.00\bar{d}_1 \cdots \bar{d}_{n-2} \cdot 10^2$

$x \to fl(x)$ $y \to fl(y)$

$$fl(x \cdot y) = fl\left(x(1+\delta_1) \cdot y(1+\delta_2)\right), \quad |\delta_i| \le \varepsilon$$

$$= xy(1+\delta_1)(1+\delta_2)(1+\delta_3)$$

relative error: input x, input y, multiplication of $fl(x) \cdot fl(y)$

$$= xy(1+\delta_1+\delta_2+\delta_3+\delta_1\delta_2+\delta_1\delta_3+\delta_2\delta_3+\delta_1\delta_2\delta_3)$$

$\delta_1+\delta_2+\delta_3 \le \varepsilon$, $\delta_1\delta_2 \le \varepsilon^2$, ε^3

$$= xy(1+\delta_4)$$

$\varepsilon = 10^{-8}$ $\varepsilon^2 = 10^{-16}$

$\varepsilon = 10^{-16}$

$|\delta_4| \le 3\varepsilon$

Subtraction, $x \in R$, $y \in R$.

$$fl(x-y) = fl\left(x(1+\delta_1) - y(1+\delta_2)\right)$$

input x input y

$$= \left(x(1+\delta_1) - y(1+\delta_2)\right)(1+\delta_3)$$

δ_3 from the subtraction

$$= (x-y)(1+?) \qquad |?| \leq$$

$$= (x + x\delta_1 - y - y\delta_2)(1+\delta_3)$$

$$= (x-y + x\delta_1 - y\delta_2)(1+\delta_3) + (x-y)(1+\delta_3)$$

$$(x-y) + (x\delta_1 - y\delta_2) + O(\varepsilon^2)$$

$$= (x-y)\left(1 + \frac{x\delta_1 - y\delta_2}{x-y}\right) + O(\varepsilon^2) + \delta_3 1$$

$$\delta_4 = \frac{x\delta_1 - y\delta_2}{x-y} + \delta_3, \quad |\delta_4| \leq ? \varepsilon$$

If x and y have different signs $x = -y$

$$|\delta_4| \leq \frac{|x||\delta_1| + |y|\delta_2}{|x|+|y|} + \varepsilon \leq \frac{|x|\varepsilon + |y|\varepsilon}{|x|+|y|} = 3\varepsilon$$

If $x-y$ is very small, the relative error can be large, very close, we may have large relative error. Cancellation of significant digits. Catastrophic

Review: machine precision $\frac{1}{2}\beta^{-n+1}$

32 bits: $n=52$.

Ex 2 $\quad fl\left(\frac{xy}{z}\right) = \frac{xy}{z}(1+\delta) \qquad |\delta| \le 5\varepsilon$ ✓

$$= \frac{x(1+\delta_1)y(1+\delta_2)(1+\delta_{xy})}{z(1+\delta_3)(1+\delta_d)}$$

$$= \frac{xy(1+\hat\delta_1)}{z(1+\hat\delta_2)} \qquad |\hat\delta_1| \le 3\varepsilon, \quad |\hat\delta_2| \le 2\varepsilon$$

$$\frac{1}{1+\hat\delta_2} = 1-\hat\delta_2+\hat\delta_2^2-\hat\delta_2^3+ \cdots +(-1)^n(\hat\delta_2)^n$$

$$\approx \frac{xy}{z}(1+\hat\delta_1)(1-\hat\delta_2)$$

$$\approx \frac{xy}{z}(1+\delta)$$

$$\frac{xy}{z} \Rightarrow \frac{xy}{z+\varepsilon\,\mathrm{sgn}(z)} \qquad (1) \qquad \frac{\nabla\varphi}{|\nabla\varphi|}$$

Aug 29-9:34 AM

How to minimize round-off errors.

$ax^2+bx+c=0. \qquad x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$

Algorithm:

A1: $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$

A2: $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \quad x_2 = \frac{c}{ax_1}, \quad b<0$

A3: $x_1 = \frac{-b-\sqrt{b^2-4ac}}{2a} \quad x_2 = \frac{c}{ax_1} \quad b>0$ ✓

Vieta's formula $x_1+x_2 = -\frac{b}{a}, \quad x_1x_2 = \frac{c}{a}$

Let $a=1$, $c=\varepsilon$, $b=2$ small number

$$x_1 = \frac{-2-\sqrt{4-4e}}{2} = -1-\sqrt{1-e} = -1-1+\frac{e}{2} = -2+\frac{e}{2} \qquad (1-e)^{\frac{1}{2}} \approx 1-\frac{1}{2}e+O(e^2)$$

$$x_2 = \frac{-2+\sqrt{4-4e}}{2} = -1+1-\frac{e}{2} = -\frac{e}{2}$$

$$\frac{x\delta_1 - y\delta_2}{x-y}$$

Aug 29-9:54 AM

How to avoid/minimize $\qquad (a+b)(a-b) = a^2-b^2$

$-b+\sqrt{b^2-4ac}$

if $b>0$, $\left(\frac{-b+\sqrt{b^2-4ac}}{-b-\sqrt{b^2-4ac}}\right)(-b-\sqrt{b^2-4ac})$ ✓

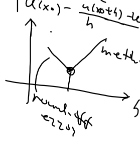
$$= \frac{(-b)^2-(b^2-4ac)}{-b-\sqrt{b^2-4ac}} = \frac{4ac}{-b-\sqrt{b^2-4ac}}$$ ✓

Ex1 $\quad 1-\cos x$

$$= \begin{cases} 1-\cos x & \text{if } |x|>\delta \\ 1-(1-\frac{x^2}{2}+\frac{x^4}{4!}) & |x|\le\delta \end{cases}$$

$$= \begin{cases} 1-\cos x \\ \frac{x^2}{2} \end{cases}$$

$$\left|u'(x_0) - \frac{u(x_0+h)-u(x_0)}{h}\right| \qquad u'(x) = \frac{u(x+h)-u(x)}{h} + O(h) \quad \text{method error}$$



method error $O(h) + \frac{\varepsilon}{h}$

The best h: $\delta = \sqrt{\varepsilon}$

Aug 29-10:10 AM

Some basic programming skills.

$\sum_{i=1}^n a_i$, $\quad S=0$ ✓ initialization.

```
for i=1:n
    S = S+a(i);
end
```

$\prod_{i=1}^n a_i$

```
P=1;
for i=1:n
    p=p*a(i);
end
```

Ax Matrix-vector multiplication.

$A\in\mathbb{R}^{m\times n}$, $x\in\mathbb{R}^{n\times 1}$ $\quad Ax\in\mathbb{R}^m \quad y=Ax$

```
for i=1:m
  y(i)=0;
  for j=1:n
    y(i)=y(i)+a(i,j)*x(j);
  end
end
```

$$i\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Aug 29-10:18 AM

$p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 x^0$

```
p=0
for i=0:n
    p = p+a(i)*x^(i);
end
```

Complexity?

Aug 29-10:25 AM

Algorithm efficiency: accuracy, fast, (FFT)
Simple, storage.
Research topic: How to obtain triple accuracy, quadruple based on the current computer system.
Evaluate a polynomial $p_n(x) = \sum_{i=0}^{n} a_i x^i$

$p_3(x) = a_3x^3 + a_2x^2 + a_1x + a_0$
$= x(a_3x^2 + a_2x + a_1) + a_0$
$= x(x(a_3x + a_2) + a_1) + a_0$

Nested algorithm

A pseudo-code

```
p = a(n)
for i = n-1, -1, 0
    p = p*x + a(i);
end
```

Complexity
+ −, n
×/÷, n

The best for sequential machines.

Aug 31-9:33 AM

Vector and matrix norms, a review.
$x \to fl(x)$ $fl(x) = x(1+\delta)$ $|\delta| \le \varepsilon$ $\frac{1}{2}\beta^{-n+1}$
$\vec{x}$ $fl(\vec{x})$

Definition of a vector norm.
Given a R^n, $R^n = \{\vec{x}, \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x_i \in R\}$
$x \in R$, $|x|$. $\vec{x} \in R^3$, $x = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$, $\|x\|_2 = \sqrt{1^2+(-1)^2+2^2} = \sqrt{6}$

A norm is a special function $f(\vec{x})$ that satisfies, similar to $|x|$ $\|x\|_\infty = 2$
(i) $f(\vec{x}) \ge 0$, $f(x) = 0$ if and only if $\vec{x} = \vec{0}$. iff
(ii) $f(\alpha\vec{x}) = |\alpha| f(\vec{x})$
(iii) $f(\vec{x}+\vec{y}) \le f(\vec{x}) + f(\vec{y})$
Then $f(\vec{x})$ is a norm in R^n, Can be written as $f(x) = \|x\|$

Aug 31-9:50 AM

Ex: 1. $f(x) = \max_{1 \le i \le n} \{|x_i|\}$

Check: 1. $f(\vec{x}) \ge 0$. If $f(x) = 0$, $\max_{1 \le i \le n} |x_i| = 0$

2. $f(\alpha\vec{x}) = \max_{1 \le i \le n} \{|\alpha x_i|\} = |\alpha| \max_{1 \le i \le n} \{|x_i|\}$
$= |\alpha| f(\vec{x})$

3. $f(\vec{x}+\vec{y}) = \max_{1 \le i \le n} \{|x_i + y_i|\} \le \max_{1 \le i \le n} \{|x_i| + |y_i|\}$
$\le \max_{1 \le i \le n} \{|x_i|\} + \max_{1 \le i \le n} \{|y_i|\}$
$= f(\vec{x}) + f(\vec{y})$

$\max\{2, 2.4, 3\}$

Aug 31-10:01 AM

Are the following vector norms?
- $f(x) = \max_{1 \le i \le n} \{|x_i|\} + 5$, $f(\vec{0}) \ne 0$
- $f(x) = \frac{\max_{1 \le i \le n} \{|x_i|\}}{x_1}$, No $f(\alpha\vec{x}) \ne |\alpha| f(\vec{x})$
- $f(x) = \left(\sum_{i=1}^{n} x_i^4\right)^{\frac{1}{4}}$, Yes.

(i) $f(\vec{x}) \ge 0$ $f(\vec{x}) = 0 \Leftrightarrow$ iff $\vec{x} = 0$
(ii) $f(\alpha\vec{x}) = \left(\sum_{i=1}^{n} (\alpha x_i)^2\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} \alpha^2 x_i^2\right)^{\frac{1}{2}}$
$= |\alpha| \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}} = |\alpha| f(\vec{x})$ $\sqrt{\alpha^2} = |\alpha|$
(iii) $f(\vec{x}+\vec{y}) \le f(\vec{x}) + f(\vec{y})$
$\sqrt{\sum_{i=1}^{n} (x_i+y_i)^2} \overset{?}{\le} \sqrt{\sum_{i=1}^{n} x_i^2} + \sqrt{\sum_{i=1}^{n} y_i^2}$ $(a+b)^2 = a^2+b^2+2ab$
$\sum_{i=1}^{n} (x_i+y_i)^2 \le \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 + 2\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}$
$\sum_{i=1}^{n} (x_i^2 + 2x_iy_i + y_i^2)$
$\sum x_i^2 + 2\sum x_iy_i + \sum y_i^2$
$\Rightarrow \sum_{i=1}^{n} x_iy_i \le \sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}$

Aug 31-10:06 AM

Cauchy-Schwartz inequality
Introduce
$g(\lambda) = \sum_{i=1}^{n} (x_i - \lambda y_i)^2 \ge 0$
$= \sum_{i=1}^{n} (x_i^2 - 2\lambda \sum_{i=1}^{n} x_i y_i + \lambda^2 y_i^2)$
$= \sum_{i=1}^{n} x_i^2 - 2\lambda \sum_{i=1}^{n} x_i y_i + \lambda^2 \sum_{i=1}^{n} y_i^2$
$= a + b\lambda + c\lambda^2$

$b^2 - 4ac \le 0$ $\frac{-b \pm \sqrt{b^2-4ac}}{2c}$

$\left(2\sum x_i y_i\right)^2 \le 4 \sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2$

$\left|\sum_{i=1}^{n} x_i y_i\right| \le \sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}$

$\sum x_i y_i \le$ $f(\vec{x}) = \sqrt{\sum_{i=1}^{n} x_i^2}$ is a vector norm
$\|x\|_2$ Euclidian Norm

Aug 31-10:17 AM