Teoria błędów

1. Błąd względny i bezwzględny:

$x = 1.23456789$ $\quad y = 10^{50} + 1$

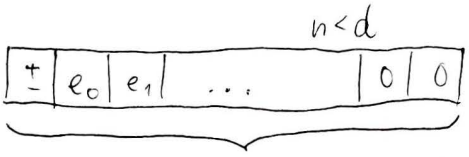
$\bar{x} = 1.23456\,79$ $\quad \bar{y} = 10^{50}$

$|x - \bar{x}| = 10^{-8}$ $\quad |y - \bar{y}| = 1$ ← błędy bezwzględne

$\left|\frac{x - \bar{x}}{x}\right| = 0.8 \cdot 10^{-8}$ $\quad \left|\frac{y - \bar{y}}{y}\right| = 10^{-50}$ ← błędy względne

2. Reprezentacja liczb:

a) całkowite: $L \in \mathbb{Z} \Rightarrow L = \pm \sum_{i=0}^{n} e_i 2^i \quad (e_i \in \{0,1\},\ e_n = 1)$



d+1 bitów na liczby całkowite ze znakiem

⚠ PROBLEM: $n > d$ ⇒ nadmiar / brak reprezentacji ⚠

Dodawanie, odejmowanie i mnożenie jest wykonalne w komputerze dokładnie, pod warunkiem, że wynik jest reprezentowalny. Problemy pojawiają się przy dzieleniu, bo wynik jest liczbą niecałkowitą.

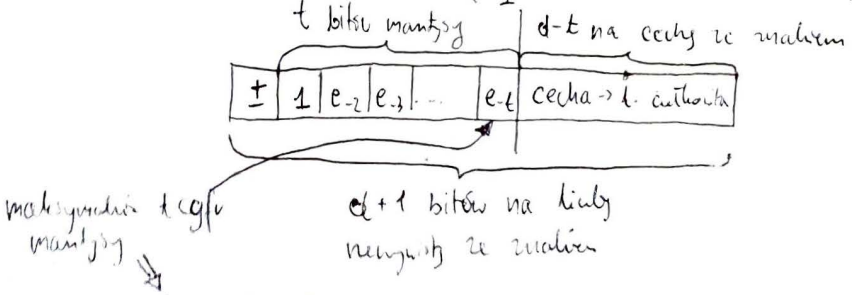
b) niecałkowite: każdą liczbę można jednoznacznie przedstawić w postaci:

$$X = s \cdot m \cdot 2^c,$$

gdzie $s \in \{+1, -1\}$ (znak), $m \in \left[\frac{1}{2}, 1\right)$ (mantysa), $c \in \mathbb{Z}$ (cecha)

⚠ PROBLEM: rozwinięcie dwójkowe jest przeważnie nieskończone ⚠

$$m = \sum_{i=1}^{\infty} e_{-i} 2^{-i} \quad (e_{-i} \in \{0,1\},\ e_{-1} = 1)$$



Rodzaje zaokrągleń:

(a) obcięcie (chopping) $\Rightarrow m \approx m_t^c := \sum_{i=1}^{t} e_{-i} 2^{-i}$ (ignorujemy cyfry od $t+1$)

(b) zaokrąglenie symetryczne (symmetric rounding) $\Rightarrow m \approx m_t^r := \sum_{i=1}^{t} e_{-i} 2^{-i} + e_{-(t+1)} 2^{-t}$

Reprezentacja zmiennoprzecinkowa liczby $x \in \mathbb{R}$ nazywamy liczby:

$$= s \cdot m \cdot 2^c$$

$$chop(x) := s \cdot m_t^c \cdot 2^c \quad \text{(obcięcie)}$$

$$rd(x) := s \cdot m_t^r \cdot 2^c \quad \text{(zaokrąglenie symetryczne)}$$

TWIERDZENIE:

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

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

Przykład:

$$\frac{2}{3} = (0.10101010\ldots)_2$$

Jeżeli $t = 24$, to $chop(x) = (0.1010\ldots10)_2$

$rd(x) = (0.1010\ldots11)_2$

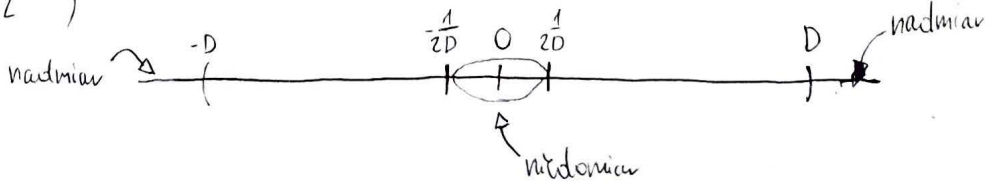
Wtedy mamy: $\left| \frac{x - chop(x)}{x} \right| = 2 \cdot 2^{-25}$

$$\left| \frac{x - rd(x)}{x} \right| = 2^{-25}$$

Jakie liczby zna komputer?

$$\frac{1}{2D} \leq |x| \leq D \quad (D = 2^{c_{max}})$$

$$\begin{cases} \text{cecha} \in [c_{min}, c_{max}] = -c_{min} = c_{max} = 2^{d-t-1} - 1 \\ m \in [\frac{1}{2}, 1) \end{cases}$$



W obliczeniach komputerowych mogą występować jedynie liczby:

$$X := \left(-D, -\frac{1}{2D}\right] \cup \{0\} \cup \left[\frac{1}{2D}, D\right)$$

W praktyce (niedomiar) zbiór dopuszczalny jest postaci:

$$X := (-D, D)$$

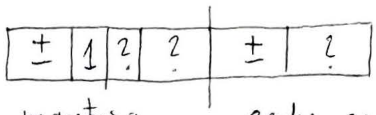
Zbiór <u>liczb zmiennopozycyjnych</u>

$$X_{fl} := \underbrace{rd(X)}_{\text{komputer zna tylko liczby z tego zbioru}}$$

Przykład:

$d = 5$

$t = 3$



$? \in \{0, 1\}$

mantysa cechę ze znakiem

więc podstawić można: $\pm\{0, \frac{1}{4}, \frac{5}{16}, \frac{3}{8}, \frac{7}{16}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4}\}$

3. Działania arytmetyczne w sensie fl.

- $x, y \in X_{fl}$
- $\circ \in \{+, -, *, /\}$
- $x \circ y \in X'$ (nie ma nadmiaru)

$$fl(x \circ y) := (x \circ y)(1 + \varepsilon_{x,y,\circ}),$$
$$|\varepsilon_{x,y,\circ}| \leq 2^{-t}$$

(t - liczba bitów na zapamiętanie mantysy)

Błąd względny:

$$\left|\frac{x \circ y - fl(x \circ y)}{x \circ y}\right| = \left|\frac{(x \circ y) - (x \circ y)(1 + \varepsilon_{x,y,\circ})}{x \circ y}\right| = \underbrace{|\varepsilon_{x,y,\circ}| \leq 2^{-t}}$$

błąd względny operacji $\circ$ wynosi co najwyżej 2^{-t}

Twierdzenie o kumulacji błędów

Jeśli $|\delta_i| \le 2^{-t}$ $(i = 1, 2, \dots, n)$, to zachodzi równość

$$\prod_{i=1}^{n} (1 + \delta_i) = 1 + \sigma_n$$

gdzie $\sigma_n = \sum_{i=1}^{n} \delta_i + O(2^{-2t})$.

Jeśli $n 2^{-t} < 2$, to $|\sigma_n| \le \gamma_n := \dfrac{n 2^{-t}}{1 - \frac{1}{2} n 2^{-t}} \approx n \cdot 2^{-t}$ (przy pewnych realistycznych założeniach)

W skrócie:

$$|\delta_i| < 2^{-t} \Rightarrow \prod_{i=1}^{n} (1 + \delta_i) = 1 + \sigma_n, \quad |\sigma_n| \lesssim n \cdot 2^{-t}$$

Przykład: Sumowanie liczb

Niech dane będą liczby $x_1, x_2, \dots, x_n \in \mathbb{R}$; obliczyć $\sum_{i=1}^{n} x_i$

PROGRAM

```
S := x_1
FOR i=2 TO n
    S := S+x_i
RETURN (S)
```

Sprawdzimy co obliczy komputer realizując podany program. Dla uproszczenia (nie zmieni to wniosku) założymy, że $x_i = rd(x_i)$ $(1 \le i \le n)$:

$$fl(S) = \Big(\dots \big(\big((x_1 + x_2)\underbrace{(1+\varepsilon_1)}_{\text{błąd pierwszego dodawania}} + x_3 \big)(1+\varepsilon_2) + x_4 \big)(1+\varepsilon_3) + \dots +$$

$$+ x_{n-1} \big)(1+\varepsilon_{n-2}) + x_n \Big)\underbrace{(1+\varepsilon_{n-1})}_{\text{błąd } (n-1)\text{szego dodawania}} =$$

$$= x_1 \overbrace{(1+\varepsilon_1)(1+\varepsilon_2)\cdots(1+\varepsilon_{n-1})}^{1+\sigma_1} +$$
$$x_2 \overbrace{(1+\varepsilon_1)(1+\varepsilon_2)\cdots(1+\varepsilon_{n-1})}^{1+\sigma_2} +$$
$$x_3 \overbrace{(1+\varepsilon_2)(1+\varepsilon_3)\cdots(1+\varepsilon_{n-1})}^{1+\sigma_3} +$$
$$\vdots$$
$$x_k \overbrace{(1+\varepsilon_{k-1})(1+\varepsilon_k)\cdots(1+\varepsilon_{n-1})}^{1+\sigma_k} +$$
$$\vdots$$
$$x_n \overbrace{(1+\varepsilon_{n-1})}^{1+\sigma_n}$$

$$= \sum_{i=1}^{n} x_i (1+\sigma_i), \text{ gdzie } \begin{cases} 1+\sigma_i = \prod_{j=i}^{n-1} (1+\varepsilon_j) & (i=2,3,\dots,n) \\ 1+\sigma_1 = 1+\sigma_2 \end{cases}$$

Z twierdzenia o kumulacji błędów wynika że:

$$|\sigma_i| \lesssim (n-i+1)\cdot 2^{-t} \quad (i=2,3,\dots,n)$$

$$|\sigma_1| \lesssim (n-1)\, 2^{-t}$$

WNIOSKI:

1° Zamiast $\sum_{i=1}^{n} x_i$ otrzymujemy (w sensie fl) $\sum_{i=1}^{n} x_i (1+\sigma_i)$:

$$\left|\frac{S-fl(S)}{S}\right| = \left|\frac{\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i(1+\sigma_i)}{\sum_{i=1}^{n} x_i}\right| = \left|\frac{\sum_{i=1}^{n} x_i \sigma_i}{\sum_{i=1}^{n} x_i}\right| \leq$$

$$\leq \frac{\sum_{i=1}^{n} |x_i||\sigma_i|}{\left|\sum_{i=1}^{n} x_i\right|} \leq \left\{\Delta := \max|\sigma_i|\right\} \leq \Delta \cdot \frac{\sum_{i=1}^{n}|x_i|}{\left|\sum_{i=1}^{n} x_i\right|}$$

↑ to może być dowolnie duże

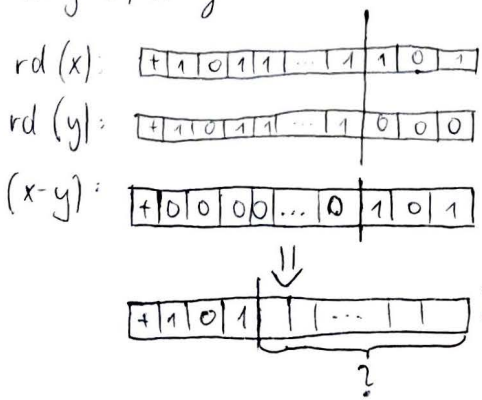
2° Jeśli $x_i > 0$ $(x_i < 0)$ $\Rightarrow \frac{\sum_{i=1}^{n}|x_i|}{\left|\sum_{i=1}^{n} x_i\right|} = 1$ $\Rightarrow$ wynik jest na poziomie $(n-1)2^{-t}$ (może dl

3° Jeśli $x_i > 0$, to liczby w sensie fl warto sumować od najmniejszej do największej (aby zminimalizować błąd).

4. Zjawisko utraty cyfr znaczących

Operacja + liczb tych samych znaków oraz * i / są uznawane za „bezpieczne". Kłopot występuje np. wtedy, gdy odejmujemy od siebie liczby tych samych znaków: (oraz odejmowanie liczb przeciwnych znaków)

$x > y > 0$, $x \approx y$



↖ musimy znormalizować liczby!

takie same cechy, bo $x \approx y$

Nie mamy co tu wpisać, to jest problem utraty cyfr znaczących.

Przykład

(1) $f(x) = \ln(x) - 1$, $x \approx e \Rightarrow f(x) \approx 0$

$f(x) = \ln\left(\frac{x}{e}\right)$

(2) $4038 \cdot \frac{1-\cos x}{x^2} = 4038 \frac{1-\cos x}{x^2} \cdot \frac{1+\cos x}{1+\cos x} = 4038 \frac{\sin^2 x}{(1+\cos x)x^2}$

drugi temat przekształcenia uzyskujemy lepszy wynik, bo unikamy problematycznego dodawania