Metody Numeryczne, część 1
studia stacjonarne

Arytmetyka zmiennopozycyjna

Krystyna Ziętak

4 październik 2018

Spis treści

William Kahan

laureat nagrody imienia Turinga
przyznanej w roku 1989 przez ACM **za IEEE standard 754**

Ten standard ustalił reguły
arytmetyki zmiennopozycyjnej w komputerze.

http://www.cs.berkeley.edu/~wkahan/

ACM: Association for Computing Machinery
IEEE: Institute of Electrical and Electronics Engineers

Liczby

- liczby całkowite: $0, 1, -1, 2, -2, \ldots$
- liczby wymierne: $\frac{1}{2} = 0.5, \ \frac{4}{3} = 1.333\ldots$
- liczby niewymierne: $\sqrt{2} = 1.41421\ldots$
- $\pi = 3.14159\ldots$
- liczba Eulera $e = \lim_{n\to\infty}\left(1 + \frac{1}{n}\right)^n = 2.718281845\ldots$
- liczby rzeczywiste

System dziesiętny - liczby całkowite

$$102 = 1 \times 10^2 + 0 \times 10^1 + 2 \times 10^0 = (102)_{10}$$

$$(2503)_{10} = 2 \times 10^3 + 5 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

$$x = x_k \times 10^k + x_{k-1} \times 10^{k-1} + \cdots + x_0 \times 10^0$$

$$x = (x_k x_{k-1} \ldots x_0)_{10}$$

$$x_i \in \{0, 1, \ldots, 9\}$$

Liczby rzeczywiste (system dziesiętny)

$$102 = 10.2 \times 10^1 = 1.02 \times 10^2 = 0.102 \times 10^3$$

$$0.0102 = 0.102 \times 10^{-1}$$

$$0.102 = 1 \times 10^{-1} + 0 \times 10^{-2} + 2 \times 10^{-3} = 0.102 \times 10^0$$

$$x = m \times 10^c$$

Założenie:
Mantysa m jest liczbą rzeczywistą z przedziału $[0.1, 1)$.
Cecha (wykładnik) c jest liczbą całkowitą.

Liczby całkowite

Podstawa systemu: 2, cyfry 0, 1

$$5 = 4 + 1 = 2^2 + 2^0 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (101)_2$$

$$13 = 8+5 = 8+4+1 = 1\times2^3+1\times2^2+0\times2^1+1\times2^0 = (1101)_2$$

$$(1011)_2 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11$$

Liczby rzeczywiste z przedziału $(0, 1)$

Podstawa systemu: 2, cyfry 0, 1

$$0.75 = 0.5 + 0.25 = 2^{-1} + 2^{-2} = 1 \times 2^{-1} + 1 \times 2^{-2} = (0.11)_2$$

$$0.625 = \frac{1}{2} + \frac{1}{8} = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (0.101)_2$$

$$(0.1011)_2 = 2^{-1} + 2^{-3} + 2^{-4} = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = \frac{11}{16}$$

Przejście od systemu dziesiętnego do dwójkowego

dodatnia liczba całkowita x

$$x = a_k 2^k + a_{k-1} 2^{k-1} + \cdots + a_1 2^1 + a_0 2^0$$

Uwaga: $2^0 = 1$

- a_0 jest resztą z dzielenia x przez 2
- Niech y będzie ilorazem z dzielenia x przez 2, czyli

$$y = a_k 2^{k-1} + a_{k-1} 2^{k-2} + \cdots + a_2 2^1 + a_1$$

- a_1 jest resztą z dzielenia y przez 2
- itd.

Przykłady

$$x = 12 = 2^3 + 2^2 = a_3 \times 2^3 + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0,$$

czyli $a_3 = a_2 = 1, \quad a_1 = a_0 = 0$

iloraz	reszta z dzielenia przez 2
12	
6	$0 = a_0$
3	$0 = a_1$
1	$1 = a_2$
	$1 = a_3$

$$x = 71 = 64 + 4 + 2 + 1 = 2^6 + 2^2 + 2^1 + 2^0$$

$$a_6 = a_2 = a_1 = a_0 = 1, \quad a_5 = a_4 = a_3 = 0$$

iloraz	reszta z dzielenia przez 2
71	
35	$1 = a_0$
17	$1 = a_1$
8	$1 = a_2$
4	$0 = a_3$
2	$0 = a_4$
1	$0 = a_5$
	$1 = a_6$

stale dzielimy przez 2

Przejście od systemu dziesiętnego do dwójkowego

x dodatnia liczba rzeczywista mniejsza od 1

$$0 < x < 1$$

$$x = a_{-1}2^{-1} + a_{-2}2^{-2} + \cdots$$

- a_{-1} jest częścią całkowitą liczby $2x$
- Niech $y = a_{-2}2^{-1} + a_{-3}2^{-2} + \cdots$ będzie częścią ułamkową liczby $2x$ (odrzucamy część całkowitą liczby $2x$).
- a_{-2} jest częścią całkowitą liczby $2y$
- itd.

$$x = \frac{5}{8} = \frac{1}{2} + \frac{1}{8} = 2^{-1} + 2^{-3} = 0.625$$

$$a_{-1} = 1 = a_{-3} = 1, \quad a_{-2} = 0$$

część całkowita	część ułamkowa
0	.625
$a_{-1} = 1$	.250
$a_{-2} = 0$	.500
$a_{-3} = 1$	.000

stale mnożymy przez 2

$$x = \frac{1}{10} = (2^{-4} + 2^{-5})(1 + 2^{-4} + 2^{-8} + \cdots)$$

część całkowita	część ułamkowa
0	.1
$a_{-1} = 0$	.2
$a_{-2} = 0$	.4
$a_{-3} = 0$	.8
$a_{-4} = 1$	.6
$a_{-5} = 1$	.2
$a_{-6} = 0$	.4
$a_{-7} = 0$	.8
$a_{-8} = 1$	.6
itd	

stale mnożymy przez 2

rozwinięcie okresowe

System dwójkowy (binarny) zmiennopozycyjny

Podstawa systemu: 2; cyfry: 0, 1

Rozpatrujemy stary model: mantysa m z przedziału $[0.5, 1)$

Do tego systemu należą liczby postaci

$$x = m \times 2^c, \quad \text{gdzie}$$

- mantysa m jest następującą liczbą
$$m = m_1 \times \frac{1}{2} + m_2 \times \frac{1}{4} + m_3 \times \frac{1}{8} + \cdots + m_t \times \frac{1}{2^t}$$
$m_i \in \{0, 1\}$ **t jest ustalone**

- cecha (wykładnik) c jest liczbą całkowitą z ustalonego przedziału $[c_{min}, c_{max}]$
$$c_{min} \leq c \leq c_{max}$$

W komputerze mantysa m:

$$m = \frac{m_1}{2} + \frac{m_2}{4} + \frac{m_3}{8} + \cdots + \frac{m_t}{2^t}$$

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

jest reprezentowana przez $m_1, \ldots, m_t$ (bity):

$$m = (0.m_1 m_2 \cdots m_t)_2$$

Mamy założenie: $0.5 \leq m < 1$. Dlatego zawsze $m_1 = 1$:

$$m = \frac{1}{2} + m_2 \times \frac{1}{4} + m_3 \times \frac{1}{8} + \cdots + m_t \times \frac{1}{2^t} =$$

Uwaga

Nie każda liczba rzeczywista x da się przedstawić w ten sposób. Dlatego niektóre liczby rzeczywiste są reprezentowane w tym systemie niedokładnie.

Przykład

Mantysa musi spełniać warunek: $0.5 \leq m < 1$

$$x = 5.5 = \frac{11}{2} = \frac{11}{16} \times 8 = 0.6875 \times 2^3$$

mantysa ma cztery bity: $m = 0.6875 = \frac{11}{16} = \frac{1}{2} + \frac{1}{8} + \frac{1}{16}$

$$m = (0.6875)_{10} = (0.1011)_2$$

cecha: $c = 3 = 2 + 1$

$$c = (3)_{10} = (11)_2$$

$$x = 0.1011E11$$

Zaokrąglenie i obcięcie mantysy

$$x = \frac{1}{10} = \frac{8}{10} \times \frac{1}{8} = \frac{4}{5} \times 2^{-3} = 0.8 \times 2^{-3}$$

Mantysa jest okresowa, ma nieskończenie wiele bitów:

$$m = \frac{4}{5} = 0.8 = 0.1100110011001100\cdots$$

Cecha:

$$c = -3$$

$$m = \frac{4}{5} = 0.8 = 0.1100110011001100\cdots$$

Cecha: $c = -3$

Mantysa z $t = 4$ bitami, obcięcie

$$m = (0.1100)_2 = (\mathbf{0.75})_{10}$$

$$\tilde{\mathbf{x}} = 0.75 \times 2^{-3} = \frac{\mathbf{3}}{\mathbf{32}} = \mathbf{0.093705}$$

Mantysa z $t = 4$ bitami, zaokr.

$$m = (0.1101)_2 = (\mathbf{0.8125})_{10}$$

$$\tilde{\mathbf{x}} = 0.8125 \times 2^{-3} = \frac{\mathbf{13}}{\mathbf{128}} = \mathbf{0.101565}$$

Przykład arytmetyki zmiennopozycyjnej dwójkowej z mantysą trzybitową

zob. N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia 2002.

$$t = 3, \quad c_{\min} = -1, \quad c_{\max} = 3$$

- mantysa z przedziału $[0.5, 1)$
- cecha - liczba całkowita z przedziału $[-1, 3]$

- **mantysy**: 0.100 0.101 0.110 0.111
- **cechy**: $-1, \quad 0, \quad 1, \quad 2, \quad 3$

wszystkie nieujemne liczby zmiennopozycyjne w tym systemie:

- 0, 0.25 0.3125 0.3750 0.4375
- 0.5 0.625 0.75 0.875
- 1.0, 1.25 1.5 1.75
- 2.0, 2.5, 3.0 3.5
- 4.0 5.0 6.0 7.0

Floating-point arithmetic - standard IEEE 754

We współczesnych komputerach stosuje się arytmetykę binarną (dwójkową) z mantysą z przedziału $[1, 2)$

Taki model przyjęto w standardzie IEEE 754.

Dlatego pierwszy bit mantysy różny od zera jest przed przecinkiem.

Liczba rzeczywista x jest reprezentowana w komputerze przez mantysę i cechę.

$$\boxed{x = m \times 2^c}$$

$$x = m \times 2^c > 0, \qquad \text{liczba rzeczywista}$$

m **mantysa**, c **cecha**

- Cecha (wykładnik) jest liczbą całkowitą
$$c_{\min} \leq c \leq c_{\max}$$
- Mantysa ma t bitów i jest liczbą rzeczywistą z przedziału
$$[1, 2).$$

Dlatego

$$m = 1 + m_1 2^{-1} + m_2 2^{-2} + \cdots + m_{t-1} 2^{1-t} =$$

$$(m_0.m_1 m_2 \cdots m_{t-1})_2$$

$$m_0 = 1, \qquad m_1, \ldots, m_{t-1} \in \{0, 1\}$$

$$x = m \times 2^c, \quad 1 \leq m < 2$$

Mantysa t-bitowa (zapis binarny):

$$m = (1.m_1 m_2 \ldots m_{t-1})_2 = (1.f)_2$$

Część ułamkowa f mantysy ma $t-1$ bitów:

$$f = m_1 \ldots m_{t-1}$$

Uwagi

- Nie zapamiętuje się jawnie w komputerze bitu m_0, bo on jest zawsze równy 1.
- W komputerze zapamiętuje się (binarnie) cechę przesuniętą:
  $\widetilde{c} = c + bias$, gdzie *bias* taki, że $\widetilde{c} > 0$

  Na przykład, dla typu *single*:
  $c_{\min} = -126$, $bias = 127$,
  więc zawsze $\widetilde{c} = c + bias > 0$.

W komputerze pamięta się cechę przesuniętą $\widetilde{c}$, część ułamkową f oraz znak liczby.

znak	$\widetilde{c}$	f

Liczby zmiennopozycyjne - typ double

$$x = m \times 2^c \qquad m = (1.f)_2$$

- część ułamkowa f mantysy ze znakiem $52 + 1$ bitów,
- cecha 11 bitów
- $\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$
  (precyzja obliczeń-*unit roundoff*)
- $c_{\min} = -1022,\ c_{\max} = 1023,\ \mathit{bias} = 1023$
- pamiętana cecha: $\widetilde{c} = c + \mathit{bias}$
- **zakres liczb**: $\pm 10^{\pm 308}$

Typ double

Mantysa ma 53 bity. Jej pierwszy bit przed przecinkiem jest zawsze równy 1. W komputerze pamiętamy tylko jej część ułamkową f, która ma 52 bity, i znak liczby (jeden bit).

Typ double

$$x = \begin{cases} \pm(1.f)_2 \times 2^{\widetilde{c}-1023} & \text{jesli } 0 < \widetilde{c} < 2047 \\ \pm 0 & \text{jesli } \widetilde{c} = 0, f = 0 \\ \pm\infty & \text{jesli } \widetilde{c} = 2047, f = 0 \\ \textbf{NaN} & \text{jesli } \widetilde{c} = 2047, f \neq 0 \\ \pm(0.f)_2 \times 2^{-1022} & \text{jesli } \widetilde{c} = 0, f \neq 0 \end{cases}$$

denormalized (subnormal) numbers

typ single

- część ułamkowa f mantysy ze znakiem 23+1 bity,
- cecha 8 bitów
- $c_{\min} = -126, \quad c_{\max} = 127, \quad bias = 127$
- **zakres liczb**: $\pm 10^{\pm 38}$
- **precyzja obliczeń**:

$$\mathbf{u} = \mathbf{2^{-24}} \approx \mathbf{5.96 \times 10^{-8}}$$

Typ single

Mantysa ma 24 bity. Jej pierwszy bit przed przecinkiem jest zawsze równy 1. W komputerze pamiętamy tylko jej część ułamkową f, która ma 23 bity, i znak liczby (jeden bit).

Zasada w standardzie IEEE 754: Każde działanie arytmetyczne daje wynik. Wynikiem może to być NaN, $\pm\infty$

NaN: **Not a Number**

Wyjątki

	przyklad	*wynik*
niedozwolona operacja	$0/0$	*NaN*
niedozwolona operacja	$0 \times \infty$	*NaN*
niedozwolona operacja	∞/∞	*NaN*
niedozwolona operacja	$\infty - \infty$	*NaN*
niedozwolona operacja	$\sqrt{-1}$	*NaN*
nadmiar		$\pm\infty$
niedomiar		*subnormal*
dzielenie $x \neq 0$ przez zero	$x/0$	$\pm\infty$

Oznaczenia:

- *fl* - floating point arithmetic
- *rd* - rounding

Epsilon maszynowy

$$\varepsilon_{\mathrm{mach}} = 2^{1-t}$$

t - liczba cyfr mantysy

Jest to odległość liczby 1 od najbliższej liczby zmiennopozycyjnej w komputerze
większej niż 1

$$1 + \varepsilon_{\mathrm{mach}} > 1$$

$\varepsilon_{\mathrm{mach}}$ - inne stosowane oznaczenie *macheps*

Wyznaczanie epsilona maszynowego, Algorytm 1

$x := 1.0$
while $1.0 + x > 1.0$ **do**
begin $\quad macheps := x$
$\qquad x := x/2.0$
end

..............................Algorytm 2

$x := 1.0$
$y := 1.0 + x$
while $y > 1.0$ **do**
begin $\quad macheps := x$
$\qquad x := x/2.0$
$\qquad y := 1.0 + x$
end

Dlaczego pierwszy algorytm może nie wyznaczyć poprawnej wartości macheps? Oblicz macheps dla double w C

Algorytm obliczania najmniejszej liczby dodatniej *eta* (znormalizowanej)

$x := 1.0$
while $x > 0$ **do**
begin

$\quad eta := x$

$\quad x := x/2.0$

end

$$\boxed{macheps \neq eta}$$

Czemu równa się *eta*?

Utrata cyfr znaczących

$$x = 0.372\underline{\mathbf{1478693}}, \qquad y = 0.372\underline{\mathbf{0230572}}$$

$$x - y = 0.0001248121 = 0.1248121 \times 10^{-3}$$

$$x - y \quad \text{po zaokrgleniu} \quad \underline{\mathbf{0.12481}} \times 10^{-3}$$

pięć cyfr znaczących

$$\mathrm{fl}(x) = 0.37215, \qquad \mathrm{fl}(y) = 0.37202$$

$$\mathrm{fl}(x) - \mathrm{fl}(y) = 0.00013 = \underline{\mathbf{0.13000}} \times 10^{-3}$$

$$\left| \frac{x - y - [\mathrm{fl}(x) - \mathrm{fl}(y)]}{x - y} \right| \approx 0.04$$

Błąd reprezentacji liczby w zmiennopozycyjnej arytmetyce dwójkowej

$$x = \pm(1 + m_1 2^{-1} + m_2 2^{-2} + \cdots)2^c, \qquad m_j \in \{0, 1\}$$

Zakładamy, że mantysa jest t-bitowa
Jeśli pierwszy odrzucony bit mantysy, czyli m_t, jest zerem, to reprezentacją liczby x jest liczba:

$$\mathrm{fl}(x) = \mathrm{rd}(x) = \pm(1 + m_1 2^{-1} + m_2 2^{-2} + \cdots + m_{t-1} 2^{-t+1})2^c$$

W przeciwnym razie, mantysę zaokrąglamy zgodnie z zasadami przyjętymi w standardzie IEEE 754. Wówczas

$$\mathrm{fl}(x) = \mathrm{rd}(x) = x(1 + \delta), \qquad |\delta| \le 2^{-t}$$

$$1.001 + 0.001 = 1.010$$

Dodawanie liczb binarnych w arytmetyce *fl*

$$x = 1 + \frac{1}{8} = 1.125 = (1.001)_2$$

$$y = \frac{1}{32} = 2^{-5} = 0.03125 = (0.00001)_2$$

suma $x + y$ obliczona w arytmetyce z mantysą 4-cyfrą równa się x, ponieważ

$$\begin{array}{l} 1.001 \\ \underline{0.00001} \\ 1.001 \end{array}$$

bez guard digit

$$x = 1.001, \qquad y = 0.0001$$

$$fl(x + y) = 1.001 + 0.000 = 1.001$$

$$x + y \quad \textit{dokladnie } 1.0011$$

po zaokrągleniu

$$rd(1.0011) = 1.010$$

z guard digit

$$fl(x + y) = 1.010$$

Niech $x = rd(x)$, $y = rd(y)$, czyli x i y są liczbami dokładnie reprezentowanymi w komputerze.

Standard IEEE 754 zakłada, że podstawowe działania na liczbach zmiennopozycyjnych komputerowych są wykonywane w komputerze tak, że

$$fl(x \Box y) = (x \Box y)(1 + \delta), \quad |\delta| \leq 2^{-t},$$

gdzie

- δ jest błędem względnym,
- $\Box$ oznacza dodawanie, odejmowanie, mnożenie, dzielenie,
- t jest liczbą cyfr mantysy.

  Na przykład, dla typu single: $t = 24$.

Przykład

Numeryczne obliczanie przybliżonej wartości pochodnej $f'(x)$

$$przybl(x,h) = \frac{f(x+h) - f(x)}{h} \approx f'(x)$$

$$blad = \left(\frac{f(x+h) - f(x)}{h}\right) - f'(x)$$

$$f(x) = \sin(x), \qquad f'(x) = \cos(x)$$
$$f'(1) = \cos(1) = 5.403023E-01,$$

$$przybl(1,h) = \frac{f(1+h)-f(1)}{h} \approx f'(1)$$
$$blad = przybl(1,h) - f'(1)$$

double precision C++

h	$przybl(1,h)$	$blad$
$1.0E-02$	$5.360860E-01$	$-4.216325E-03$
$1.0E-07$	$5.403023E-01$	$-4.182769E-08$
$1.0E-08$	$5.403023E-01$	$-2.969885E-09$
$1.0E-09$	$5.403024E-01$	$+5.254127E-08$
$1.0E-10$	$5.403022E-01$	$-5.848104E-08$
$1.0E-11$	$5.403011E-01$	$-1.1668704E-06$
$1.0E-15$	$5.551115E-01$	$+1.480921E-02$
$1.0E-20$	$0.000000E+00$	$-5.4403023E-01$

Jak obliczać $\mathbf{a}^2 - \mathbf{b}^2$?

$$\mathbf{c} = (\mathbf{a} - \mathbf{b})(\mathbf{a} + \mathbf{b}) = \mathbf{a}^2 - \mathbf{b}^2$$

Jak obliczać pierwiastki trójmianu kwadratowego?

$$x^2 + px + q = 0$$

$$\Delta = p^2 - 4q$$

Analiza błędów zaokrągleń

$$c = (a - b)(a + b) = a^2 - b^2$$

Algorytm I

$$fl((a - b)(a + b)) =$$
$$= (a - b)(1 + \delta_1)(a + b)(1 + \delta_2)(1 + \delta_3) =$$
$$= (a^2 - b^2)(1 + \beta)$$

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

$$|\delta_i| \leq 2^{-t}$$

$$1 + \beta = (1 + \delta_1)(1 + \delta_2)(1 + \delta_3) =$$
$$1 + \delta_1 + \delta_2 + \delta_3 + \delta_1\delta_2 + \delta_1\delta_3 + \delta_2\delta_3 + \delta_1\delta_2\delta_3$$
$$\approx 1 + \delta_1 + \delta_2 + \delta_3$$

W przybliżeniu:

$$|\beta| \leq 3 \times 2^{-t}$$

$$|\delta_i| \leq 2^{-t}$$

Algorytm II

$$\mathrm{fl}(\mathbf{a}^2 - \mathbf{b}^2) =$$

$$= \Big[(\mathbf{a} \times \mathbf{a})(1+\delta_1) - (\mathbf{b} \times \mathbf{b})(1+\delta_2)\Big](1+\delta_3) =$$

$$= (\mathbf{a}^2 - \mathbf{b}^2)(1+\gamma)(1+\delta_3)$$

$$|\delta_i| \leq 2^{-t}$$

$$|\gamma| \leq \frac{1+\frac{b^2}{a^2}}{|1-\frac{b^2}{a^2}|} \times 2^{-t}$$

Oszacowanie duże, jeśli $\frac{b^2}{a^2} \approx 1$ (*bliskie* 1). Grozi utrata cyfr znaczących.

$$x^2 + px + q = 0$$

$$\Delta = p^2 - 4q$$

$$x_1 = \frac{-p + \sqrt{\Delta}}{2} = \frac{(-p + \sqrt{\Delta})(-p - \sqrt{\Delta})}{2(-p - \sqrt{\Delta})} = \frac{q}{x_2}$$

$$x_2 = \frac{-p - \sqrt{\Delta}}{2}$$

$$x_1 = \begin{cases} \frac{-p+\sqrt{\Delta}}{2} & \text{jesli } p \leq 0 \\ \frac{2q}{-p-\sqrt{\Delta}} & \text{jesli } p > 0 \end{cases}$$

- Zmiennopozycyjna reprezentacja liczby rzeczywistej - mantysa i cecha (wykładnik)
- Przejście od systemu dziesiętnego do dwójkowego (binarnego)
- Standard IEEE 754
- Algorytm wyznaczania epsilona maszynowego
- Utrata cyfr znaczących
- Arytmetyka zmiennopozycyjna - analiza błędów zaokrągleń
- Przykłady jak wykonywać obliczenia w komputerze

Literatura

1. D. Kincaid, W. Cheney, *Analiza numeryczna*, WNT, Warszawa 2006.
2. N. J. Higham, *Accuracy and stability of numerical algorithms*, SIAM, Philadelphia 2002.