

I Wykład: 15 X 10r

I–1. Wprowadzenie

Rysunki są wykonane przy pomocy *MatLab'a R12* firmy *The MathWorks, Inc.*

1

I-1.1. Liczby całkowite**I-1.2. Liczby rzeczywiste**

Liczba „maszynowa” [2, ss. 9–19]:

0	100 0010	1011 0011 0000 0100 0000 0000
---	----------	-------------------------------

1	7 bitów	24 bity
<i>s</i>	<i>e</i>	<i>m</i>

Tablica I-1. „Liczba maszynowa” w komputerach IBM 370

reprezentuje następującą liczbę „matematyczną” zapisaną w postaci dziesiętnej

$$+ \left(\left(\frac{1}{2} \right)^1 + \left(\frac{1}{2} \right)^3 + \left(\frac{1}{2} \right)^4 + \left(\frac{1}{2} \right)^7 + \left(\frac{1}{2} \right)^8 + \left(\frac{1}{2} \right)^{14} \right) \times 16^{66-64} = 179.015625$$

Takie liczby rzeczywiste pojedynczej precyzji były używane w komputerach IBM 370. Jest to przykład [6, ss. 47–56] przedstawienia liczby w postaci:

$$x = (-1)^0 \times (11 \cdot 16^{-1} + 3 \cdot 16^{-2} + 0 \cdot 16^{-3} + 4 \cdot 16^{-4} + 0 \cdot 16^{-5} + 0 \cdot 16^{-6}) \times 16^2$$

lub szesnastkowo¹

$$x = (0.b30400)_{16} + 2 = (0.179015625)_{10} + 3$$

ogólnie²:

$$x = (-1)^s \cdot (0.a_1a_2 \dots a_t)_\beta \cdot \beta^e = (-1)^s \cdot m \cdot \beta^e$$

gdzie

- $s \in \{0, 1\}$ – bit znaku,
- t – ilość znaczących cyfr 24 bitowej mantysy $m = (0.a_1a_2 \dots a_t)_\beta$ z cyframi $0 \leq a_i \leq \beta - 1$, $i = 1, 2, \dots, t$, zapisanej przy podstawie β . W przykładzie $t = 6$ i $\beta = 16$. Mantysa m jest ułamkiem $0 \leq m \leq 1 - \beta^{-t}$ takim, że $\beta^t m$ jest dowolną liczbą naturalną spełniającą $0 \leq \beta^t m \leq \beta^t - 1$.
- e – cecha. W przykładzie 7 bitów, liczba z przedziału $e \in \{-64, -63, \dots, +63\}$. W przedstawieniu „maszynowym liczby” przyjęto umowę:

$$e = 100\,0010_2 - 100\,0000_2 = 66_{10} - 64_{10}.$$

Przy wyliczeniu cechy e wystąpiła liczba -64 , która „przesunęła” 66 do uzyskania poprawnej wartości $e = 2$. Liczba ta nosi nazwę bias – odchylenie i służy zaoszczędzeniu jednego bitu jaki trzeba by było przeznaczyć na znak cechy.

¹ Za [7, ss. 15] i [1, ss. 88] przyjmujemy oznaczenie:

$$(0.a_1a_2 \dots a_t)_\beta + e = \beta^e \sum_{i=1}^t a_i \beta^{-i} \quad \text{gdzie} \quad a_i \in \{0, 1, \dots, \beta - 1\}.$$

² W [4, t. 2, ss. 207–352] przyjęto zapis

$$(e, f) = f \times b^{e-q}$$

gdzie q – nadmiar.

Tak więc matematycznie liczby rzeczywiste jakimi posługujemy się w komputerze (notacja za [6, ss. 47–56]) można zapisać w postaci zbioru

$$\mathbb{F}(\beta, t, L, U) = \{0\} \cup \{x \in \mathbb{R} : x = (-1)^s \beta^e \sum_{i=1}^t a_i \beta^{-i}\}, \quad L \leq e \leq U. \quad (\text{I-1.1})$$

stanowią skończony podzbiór zbioru liczb rzeczywistych $\mathbb{R}$. W naszym liczbowym przykładzie dla komputera IBM 370 liczby rzeczywiste te tworzą zbiór:

$$\mathbb{F}(16, 6, -64, 63).$$

W standartowych językach programowania spotykamy się z co najmniej dwoma rodzajami liczb rzeczywistych: pojedynczej i podwójnej precyzji. Graficznie zajętość przez nie pamięci maszyny można przedstawić w następujący sposób ([6, ss. 47], [3, ss. 42], [1, ss. 90]):

1	8 bitów	23 bity
s	e	m

1	11 bitów	52 bity
s	e	m

Tablica I-2. Liczby pojedynczej i podwójnej precyzji w standardzie IEEE.

W przyjętych oznaczeniach odpowiada to następującym parametrom zbioru liczb $\mathbb{F}(2, t, L, U)$.

precyzja	zajętość	t	L	U	ε_M
pojedyncza	32	$23 + 1$	-126	127	$2^{-24} \approx 5.96 \cdot 10^{-8}$
podwójna	64	$52 + 1$	-1022	1023	$2^{-53} \approx 1.11 \cdot 10^{-16}$

Tablica I-3. Parametry opisujące liczby pojedynczej i podwójnej precyzji.

W celu uniknięcia kłopotów rachunkowych dalsze nasze rozważania ograniczymy do liczb sześciobitowych $\mathbb{F}(2, 3, -1, 2)$ (za [1, ss. 88], [6, ss. 48]). Z przedstawienia (I-1.1) liczbę $(0.5)_{10}$ można zapisać w postaci maszynowej następująco:

$$(0.100)_2 + 0 = (0.010)_2 + 1 = (0.001)_2 + 2 = (0.5)_{10}.$$

Dla liczb z $\mathbb{F}(2, 3, -1, 2)$ nie ma więc jednoznaczności przedstawienia w postaci pary:

$$(m, e) \quad 0 \leq m < 1, \quad L \leq e \leq U.$$

Poprawiono więc definicję (I-1.1) w następujący sposób ($L \leq e \leq U$):

$$\mathbb{F}(\beta, t, L, U) = \{0\} \cup \{x \in \mathbb{R} : x = (-1)^s \beta^e \sum_{i=1}^t a_i \beta^{-i}, \quad a_1 \neq 0\}. \quad (\text{I-1.2})$$

Skutek jest taki, że co prawda zapis w postaci pary (m, e) jest jednoznaczny, ale uzyskano to kosztem odrzucenia, poza zerem liczb $|x| < \beta^{L-1}$. W przypadku naszych liczb sześciobitowych są to:

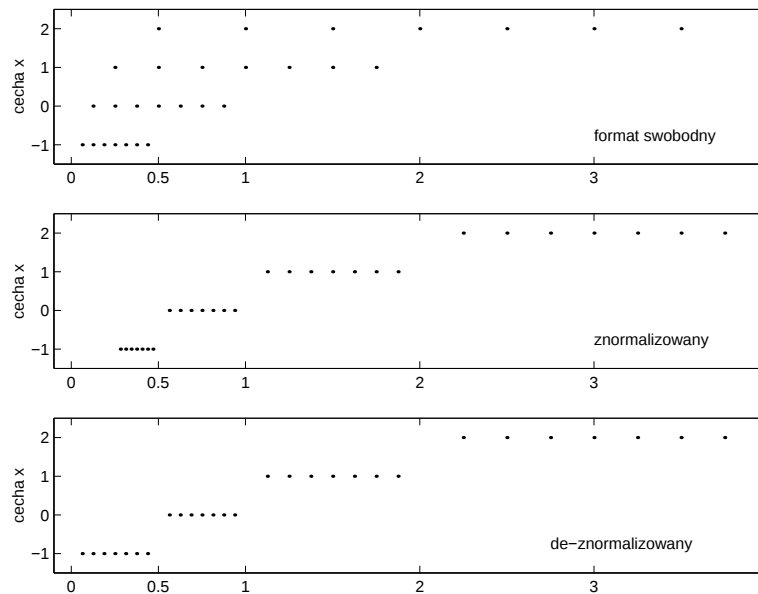
$$\pm(0.001)_2 - 1, \pm(0.010)_2 - 1, \pm(0.011)_2 - 1.$$

Zapis liczby w definicji (I-1.2) nazywa się znormalizowanym. W szczególnym przypadku podstawy $\beta = 2$ można zastosować jeszcze jeden „chwyt” wynikający z oczywistego faktu, że jeżeli $a_1 \in \{0, 1\}$ i $a_1 \neq 0$ to $a_1 = 1$ i informacja o tym jest zapisana na jednym bicie. W ten sposób w zapisie maszynowym liczb znormalizowanych przy podstawie $\beta = 2$ bit a_1 można pominąć („ukryć”) a zwolnione miejsce wykorzystać dla dodatkowego bitu zwiększając tym samym długość mantysy o jeden³.

$$\mathbb{F}(2, t+1, L, U) = \{0\} \cup \{x \in \mathbb{R} : x = (-1)^s 2^e ((0.1)_2 + \sum_{i=2}^{t+1} a_i 2^{-i})\} \quad (\text{I-1.3})$$

tnz.

$$x \in \mathbb{F}(2, t+1, L, U) \iff x = \pm(0.1 a_2 a_3 \dots a_{t+1})_2 + e.$$



Rysunek I-1. Rozkład dodatnich liczb sześciobitowych dla różnych formatów.

Dla naszych sześciobitowców (które są formalnie „znormalizowanymi siedmiobitowcami”) na Rys. I-1, str. 4 widać jakie otrzymaliśmy liczby. Z rysunku tego wynika, że normalizacja liczb komputerowych doprowadziła do powstania wokół zera luki $(-\beta^{L-1}, \beta^{L-1})$, gdzie jedyną liczbą z naszej reprezentacji jest 0. Aby zlikwidować powstałą lukę proces „normalizacji liczb” psuje się w następujący sposób

$$\mathbb{F}_D(2, t+1, L, U) = \{0\} \cup \{x \in \mathbb{R} : x = (-1)^s 2^e ((0.1)_2 + \sum_{i=2}^{t+1} a_i 2^{-i})\} \cup$$

³ Postąpiliśmy tak zgodnie z [6, ss. 48] natomiast w [3, ss. 41–46] przyjęto następującą interpretację „ukrytego” bitu

$$x = \pm(1.a_1 a_2 \dots a_t)_2 + e.$$

$$\cup \{x \in \mathbb{R} : x = (-1)^s 2^L \sum_{i=1}^t a_i 2^{-i}\} \quad (\text{I-1.4})$$

tn.

$$x \in \mathbb{F}_D(2, t+1, L, U) \iff$$

$$\iff x = \pm(0.1 a_2 a_3 \dots a_{t+1})_2 + e \quad \vee \quad x = \pm(0.a_1 a_2 \dots a_t)_2 + L,$$

gdzie $L < e \leq U$ („ostra” nierówność jest tu istotna !). Dla liczb $|x| < 2^L$ odrzucono założenie $a_1 \neq 0$. Innymi słowy nastąpiła ich de-normalizacja i tak w [6, ss. 48] liczby ze zbioru $\mathbb{F}_D(2, t+1, L, U)$ nazwano de-znormalizowanymi⁴. W wyniku tego procesu nie utracono jednoznaczności przedstawienia w postaci pary (m, e) i równocześnie zlikwidowano lukę wokół zera.

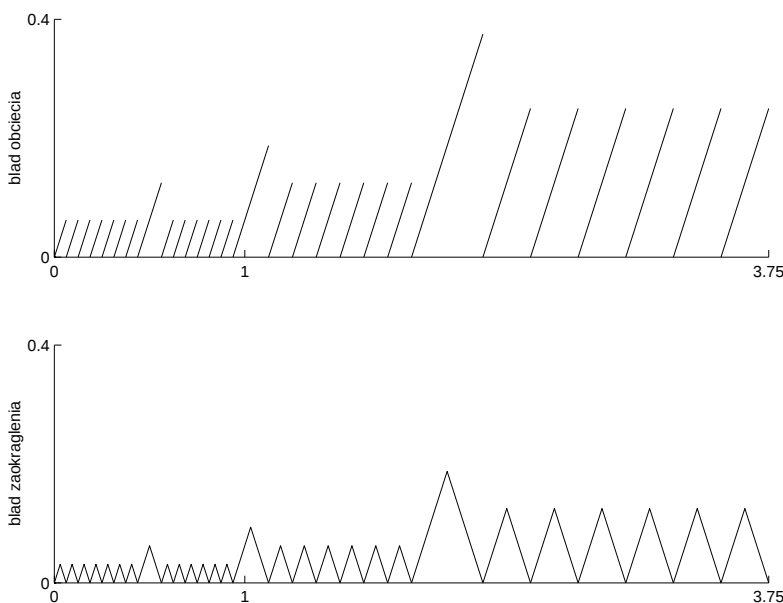
W przypadku naszego sześciobitowego układu z Rys. I-1, str. 4 widać, że procedura de-normalizacji polega na takim samym rozłożeniu liczb na odcinku $(0, 2^{L-1})$ co na $(2^{L-1}, 2^L)$. Zakres zmienności $x \in \mathbb{F}_D(2, t+1, L, U)$, $x \neq 0$

$$2^{L-t} = x_{\min} \leq |x| \leq x_{\max} = 2^U (1 - 2^{-t-1})$$

i wartości krańcowe są osiągnięte⁵.

I-1.3. Przedstawienie liczb w komputerze

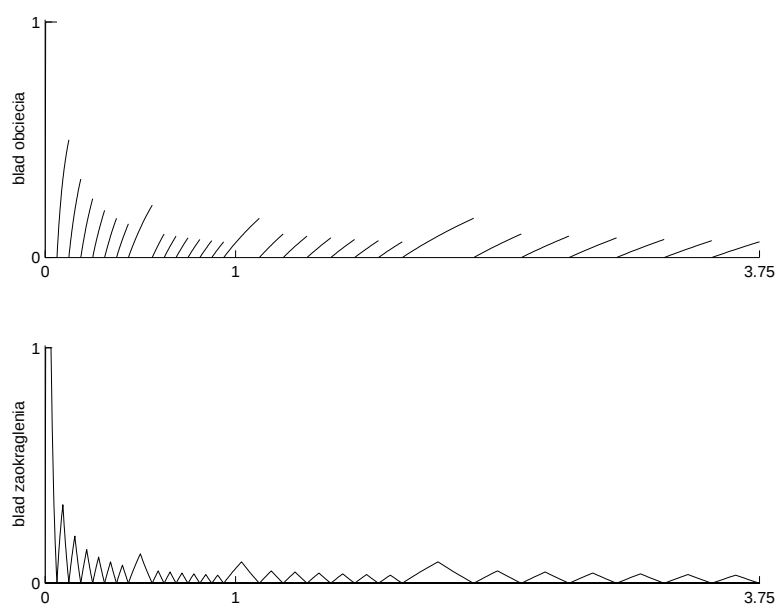
Wiadomo [5, ss. 5], że wartością przybliżoną danej wielkości jest dowolna liczba.



Rysunek I-2. Błąd bezwzględny dla $\mathbb{F}_D(2, 3+1, -1, 2)$.

⁴ de-normalized numbers

⁵ W [4, t. 2, ss. 207–352] wprowadzono jednostkę: $1\text{ulp} \stackrel{\text{def}}{=} \beta^{1-t}$ „ulp” – „unit in the last place”. W [3, ss. 47] nazwano β^{1-t} – unit roundoff error. Trzeba tu jednak wziąć pod uwagę sposób zaokrąglania liczb.

Rysunek I-3. Błąd względny dla $\mathbb{F}_D(2, 3 + 1, -1, 2)$.

Literatura

- [1] A. Björck and G. Dahlquist, Numerical mathematics and scientific computation, SIAM, Philadelphia, 2003,
Wersję roboczą polecił A. Szustalewicz. Jest ona dostępna pod adresem <http://www.mai.liu.se/~akbjjo/NMbook.html>.
- [2] R.L. Burden, J.D. Faires, and Reynolds A.C., Numerical analysis, Prindle, Weber & Schmidt, Boston, Massachusetts, 1981.
- [3] D. Kincaid and W. Cheney, Numerical analysis, Brooks, USA, 1996,
Książka jest w bibliotece Instytutu Informatyki Uniwersytetu Wrocławskiego.
- [4] D.E. Knuth, Sztuka programowania, WNT, Warszawa, 2002,
Zawsze warto tam zaglądnąć.
- [5] St. Paszkowski, Zbiór zadań z teorii metod numerycznych, PWN, Warszawa, 1969,
Biały kruk bibliofilski.
- [6] A. Quarteroni, R. Sacco, and F. Saleri, Numerical mathematics, Springer, 2000.
- [7] A. Ralston, Wstęp do analizy numerycznej, PWN, Warszawa, 1983,
Tłumaczył R. Zuber, wydanie polskie opracował St. Paszkowski. Do wydania polskiego jest dedykacja autora.

Wrocław, dnia 15 listopada 2010

(-) Krzysztof Tabisz
`tabisz@math.uni.wroc.pl`