Arytmetyka zmiennopozycyjna a dokładność obliczeń

Leszek PLASKOTA

W systemie dziesiętnym liczby całkowite zapisujemy jako skończone ciągi cyfr, czyli symboli ze zbioru $\{0,1,2,3,4,5,6,7,8,9\}$.

Jeśli liczba jest ujemna, to taki ciąg poprzedzamy minusem.

Liczby wymierne piszemy zwykle w postaci ułamka p/q, gdzie p jest liczbą całkowitą, a q – naturalną. Jest to wskazówka, że p/q to wynik dzielenia p całości na q równych części. Podobnie możemy patrzeć na „szkolny" zapis liczb niewymiernych: $\sqrt{2}$ np. to nic innego, jak zapis operacji pierwiastkowania. Przedstawia on liczbę, której kwadrat jest równy 2. Podobnie, pisząc π, mamy na myśli stosunek obwodu koła do jego średnicy.

Zapis liczb jako wyników pewnych operacji czy też zapis symboliczny ma niewątpliwe zalety, ale też pewne wady. Główną zaletą jest to, że liczba jest zawsze dokładnie zdefiniowana. Nie znaczy to jednak (i to jest wada), że znamy *wartość numeryczną* danej liczby, czyli kolejne cyfry jej rozwinięcia dziesiętnego.

Liczby niewymierne (i niektóre liczby wymierne) mają rozwinięcia dziesiętne nieskończone. Zapis dziesiętny *musimy* więc gdzieś „uciąć" i zadowolić się pewnym *przybliżeniem*. Jeśli zapiszemy tylko t cyfr po przecinku i ostatnią zaokrąglimy (tzn. zwiększymy o jeden, gdy po niej stoi jedna z cyfr $5,6,7,8,9$), to otrzymamy reprezentację z dokładnością nie mniejszą niż $\delta = 0{,}5 \cdot 10^{-t}$.

Mamy na przykład

$$\frac{1}{1024} = 0{,}0009765625,$$
$$\sqrt{2} = 1{,}4142135\ldots,$$
$$\pi = 3{,}14159265\ldots.$$

Jeśli np. chcemy mieć $t = 6$ cyfr po przecinku, to liczba $1/1024$ będzie reprezentowana przez $0{,}000977$, a π przez $3{,}141593$.

Opisany sposób to tzw. *reprezentacja stałoprzecinkowa*. Jej błąd bezwzględny nie przekracza *stałej* δ, niezależnie od wielkości liczby. Jest to własność niepożądana – w konsekwencji liczby bardzo małe są reprezentowane przez zero! Choć w świecie olbrzymów centymetr długości niewiele znaczy, to w świecie liliputów ten sam centymetr może oznaczać bardzo dużo. Mówiąc mniej obrazowo, wolelibyśmy reprezentować „małe" liczby z „dużą" dokładnością, nawet za cenę zmniejszenia dokładności „dużych" liczb.

Postulat ten spełnia *reprezentacja zmiennoprzecinkowa*, którą teraz opiszemy. Załóżmy najpierw, że $x > 0$, oraz że c_i są kolejnymi cyframi rozwinięcia dziesiętnego liczby x, tzn.

$$x = c_k\, 10^k + \cdots + c_1\, 10 + c_0 + \frac{c_{-1}}{10} + \frac{c_{-2}}{10^2} + \cdots,$$

przy czym $c_k \neq 0$. Wtedy reprezentacja zmiennoprzecinkowa liczby x z uwzględnieniem t cyfr znaczących jest równa

$$\mathrm{rd}(x) = \begin{cases} \displaystyle\sum_{j=k-t+1}^{k} c_j 10^j, & \text{gdy } c_{k-t} \leq 4, \\ \displaystyle\sum_{j=k-t+1}^{k} c_j 10^j + 10^{k-t+1}, & \text{gdy } c_{k-t} \geq 5. \end{cases}$$

(Skrót rd pochodzi od angielskiego *round-off* – zaokrąglać.) Dla $x < 0$ kładziemy $\mathrm{rd}(x) = -\mathrm{rd}(-x)$. Dodatkowo przyjmujemy, że liczba $x = 0$ jest reprezentowana dokładnie: $\mathrm{rd}(0) = 0$. Zauważmy, że w reprezentacji zmiennoprzecinkowej interesuje nas nie tyle t cyfr po przecinku, co t cyfr „najważniejszych".

Często posługujemy się jednolitym zapisem $\mathrm{rd}(x) = z \cdot m_t \cdot 10^c$, gdzie $z \in \{-1,1\}$ jest *znakiem*, m_t liczbą z przedziału $[0{,}1;\ 1{,}0)$ zwaną *mantysą*, a c liczbą całkowitą zwaną *cechą* liczby.

W ten sposób można jednoznacznie zapisać każdą liczbę zmiennoprzecinkową różną od zera. Jeśli interesują nas np. $t = 4$ cyfry znaczące, to mamy

$$\mathrm{rd}\left(\frac{1}{1024}\right) = 0{,}9766 \cdot 10^{-3},$$
$$\mathrm{rd}\left(\sqrt{2}\right) = 0{,}1414 \cdot 10^{1},$$
$$\mathrm{rd}\left(-\pi\right) = -0{,}3142 \cdot 10^{1}.$$

Piszemy również $0{,}1234 \cdot 10^{-5}$ zamiast $0{,}000001234$ i $0{,}5678 \cdot 10^{9}$ zamiast 567800000.

Bardzo wygodny zapis zmiennopozycyjny stosuje się współcześnie we wszelkich obliczeniach automatycznych. W szczególności, w ten sposób reprezentuje się liczby rzeczywiste w *maszynach cyfrowych*, przy czym tam t zwykle waha się między 8 a 16. Mantysy i cechy używamy też często w obliczeniach „ręcznych", by zaznaczyć rząd wielkości liczby i nie przepisywać wielu zer, gdy liczba jest bardzo mała lub bardzo duża.

Jaka jest dokładność reprezentacji zmiennoprzecinkowej? Czytelnik sprawdzi, że $\mathrm{rd}(x) = x(1+\varepsilon)$, gdzie ε jest liczbą „małą", $|\varepsilon| \leq 5 \cdot 10^{-t}$. Zmiennopozycyjna reprezentacja liczb rzeczywistych daje więc *błąd względny* nie większy niż

$$\nu = 5 \cdot 10^{-t}.$$

Wielkość ν nazywamy *dokładnością reprezentacji*.

Zajmiemy się teraz rachowaniem na *liczbach zmiennoprzecinkowych*, czyli na elementach zbioru $\overline{\mathbb{R}} = \{\mathrm{rd}(x) : x \in \mathbb{R}\}$. Zauważmy, że jeśli nawet $x, y \in \overline{\mathbb{R}}$, tzn. $\mathrm{rd}(x) = x$ i $\mathrm{rd}(y) = y$, to wynik operacji arytmetycznej na x i y nie musi być wcale liczbą zmiennoprzecinkową.

Jeśli np. $t = 3$ oraz $x = y = 0{,}638$, to

$$x * y = 0{,}407044 \neq \mathrm{rd}(x * y) = 0{,}407.$$

Podobnie,

$$x + y = 1{,}276 \neq \mathrm{rd}(x + y) = 0{,}128 \cdot 10^{1}.$$

Oto bardziej drastyczny przykład. Dla $t = 3$ niech $x = 123$ i $y = 0{,}456$. Wtedy $x + y = 123{,}456$, ale $\mathrm{rd}(x + y) = 123 = x$. Efekt jest taki, jakbyśmy dodali zero!

Aby wykonać jedno z działań arytmetycznych na dwóch liczbach zmiennoprzecinkowych, najpierw wykonujemy owo działanie dokładnie, a potem wynik reprezentujemy zmiennoprzecinkowo. Zbiór $\overline{\mathbb{R}}$ liczb zmiennoprzecinkowych z tak zdefiniowanymi operacjami arytmetycznymi będziemy nazywać *arytmetyką zmiennoprzecinkową* i oznaczać przez fl (od angielskiej nazwy *floating point arithmetic*). Jeśli więc $\Box \in \{+, -, *, /\}$, to wówczas

$$\mathrm{fl}(x \Box y) = \mathrm{rd}(x \Box y) = (x \Box y)(1 + \varepsilon), \quad \text{gdzie } |\varepsilon| \leq \nu.$$

Zatem, błąd względny pojedynczej operacji arytmetycznej nie przekracza ν. A jak jest przy obliczaniu w fl bardziej skomplikowanych wyrażeń? Rozważmy najpierw mnożenie trzech liczb $x, y, z \in \overline{\mathbb{R}}$. Ponieważ wykonanie dowolnej operacji powoduje powstanie błędu względnego na poziomie ν, więc

$$\begin{aligned}\mathrm{fl}(x * y * z) &= (((x * y)(1 + \varepsilon_1)) * z)(1 + \varepsilon_2) = \\ &= x * y * z(1 + \varepsilon),\end{aligned}$$

gdzie $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2$. Ponieważ $|\varepsilon_1|, |\varepsilon_2| \leq \nu$, więc $|\varepsilon| \leq 2\nu + \nu^2$. Dla dużych t liczba ν^2 jest dużo mniejsza niż ν (np. jeśli $t = 8$, to $\nu = 5 \cdot 10^{-8}$, a $\nu^2 = 2{,}5 \cdot 10^{-15}$). Możemy więc powiedzieć, że przy mnożeniu trzech liczb otrzymujemy w fl wynik, którego błąd względny jest nie większy niż 2ν. Podobnie, mnożenie n liczb zmiennoprzecinkowych powoduje błąd względny na poziomie nie większym niż $(n-1)\nu$ i jest operacją „bezpieczną" dla n niezbyt wielkich w porównaniu z $1/\nu$.

Zobaczmy teraz, co dzieje się, gdy dodajemy trzy liczby x, y, z. Mamy

$$\begin{aligned}\mathrm{fl}(x + y + z) &= (\mathrm{fl}(x + y) + z)(1 + \varepsilon_2) = \\ &= ((x + y)(1 + \varepsilon_1) + z)(1 + \varepsilon_2).\end{aligned}$$

Stąd, pamiętając o nierównościach $|\varepsilon_i| \leq \nu$, łatwo otrzymujemy

$$|\mathrm{fl}(x + y + z) - (x + y + z)| \leq (2\nu + \nu^2)(|x| + |y| + |z|).$$

Błąd względny wyniku można więc mniej więcej oszacować jako

$$\frac{|\mathrm{fl}(x + y + z) - (x + y + z)|}{|x + y + z|} \leq 2\nu \frac{|x| + |y| + |z|}{|x + y + z|}.$$

Zatem, błąd względny reprezentacji ν może być „wzmocniony" współczynnikiem $K = 2(|x| + |y| + |z|)/|x + y + z|$.

Jeśli x, y i z mają ten sam znak, to $K = 2$ i oszacowanie jest podobne do tego przy mnożeniu. Jednak w ogólności K może być dowolnie duże – np. wtedy, gdy suma $x + y + z$ jest bliska zeru, ale poszczególne składniki mają duże wartości bezwzględne. Zatem dodawanie liczb o różnych znakach *może* powodować duży błąd względny wyniku. Pamiętajmy o tym zawsze, gdy wykonujemy obliczenia „ręcznie", zaokrąglając wyniki częściowe.

Oto patologiczny przykład. Niech $t = 4$, $x = y = 0{,}5678$, $z = -1{,}135$. Wtedy $x + y + z = 0{,}6 \cdot 10^{-3}$, ale $\mathrm{fl}(x + y + z) = 0$. Błąd względny wyniku jest nieskończony!

Ostatnie stwierdzenie brzmi pesymistycznie. Czyżby arytmetykę zmiennopozycyjną można było wyrzucić do kosza? Oczywiście nie. Powyższe oszacowania błędów są zwykle zawyżone (choć bywają osiągane!). W praktyce błędy są zwykle dużo mniejsze, a czasem nawet wzajemnie się redukują. Poza tym, często możemy zabezpieczyć się przed nadmiernym błędem nieco modyfikując arytmetykę. Możemy np. wyniki częściowe otrzymywać w arytmetyce o wyższej precyzji, a dopiero wynik końcowy zaokrąglić do precyzji nas interesującej. Proszę sprawdzić, że gdybyśmy w przykładzie z dodawaniem trzech liczb zastosowali arytmetykę z $t = 5$ zamiast $t = 4$, to otrzymalibyśmy wynik dokładny. Ten prosty pomysł, godny polecenia przy obliczeniach „ręcznych", jest często stosowany w obliczeniach na maszynach cyfrowych i prowadzi do tzw. arytmetyki rozszerzonej.

Innym sposobem uniknięcia nadmiernych błędów jest modyfikacja samego sposobu obliczeń, czyli *algorytmu*. Oto przykład. Wartość wyrażenia $f(x, y) = x^2 - y^2$ dla danych $x, y \in \overline{\mathbb{R}}$ możemy obliczyć dwojako: stosując wzór $f(x, y) = x * x - y * y$, albo $f(x, y) = (x - y) * (x + y)$. Który z nich jest lepszy? W pierwszym przypadku mamy

$$\mathrm{fl}(x * x - y * y) = (x^2(1 + \varepsilon_1) - y^2(1 + \varepsilon_2))(1 + \varepsilon_3),$$

gdzie $\varepsilon_1, \varepsilon_2$ są błędami powstałymi przy podnoszeniu x i y do kwadratu, a ε_3 jest błędem powstałym z odejmowania. Błąd względny, równy

$$\left|\varepsilon_3 + (1 + \varepsilon_3)\frac{\varepsilon_1 x^2 - \varepsilon_2 y^2}{x^2 - y^2}\right|,$$

może być bardzo duży (np. dla $|x| \approx |y|$ i ε_1, ε_2 różnych znaków). Natomiast przy drugim sposobie liczenia błąd względny na ogół nie przekracza 3ν.

Dla ilustracji podamy jeszcze przykład numeryczny. Niech $t = 3$, $x = 0{,}567$ i $y = 0{,}566$. Wtedy

$$x^2 - y^2 = 0{,}1133 \cdot 10^{-2},$$

$$\mathrm{fl}(x * x - y * y) = 0{,}1 \cdot 10^{-2},$$

$$\mathrm{fl}((x - y) * (x + y)) = 0{,}113 \cdot 10^{-2}.$$

Błąd względny w drugim sposobie liczenia jest więc prawie 50 razy mniejszy niż w pierwszym.

W praktyce, zwłaszcza związanej z obliczeniami na maszynach cyfrowych, spotykamy dużo większe problemy. Arytmetyka maszyny cyfrowej jest w rzeczywistości również bardziej skomplikowana. Ponadto, zanim zaczniemy stosować jakiś algorytm, musimy najpierw przeanalizować liczbę operacji arytmetycznych koniecznych do jego wykonania, a także jego *własności numeryczne* – czyli reakcje na obliczenia prowadzone w arytmetyce zmiennoprzecinkowej. Zajmuje się tym m.in. gałąź matematyki zwana *analizą numeryczną*.