Analiza numeryczna

Stanisław Lewanowicz

Październik 2007 r.

Podstawowe pojęcia teorii błędów w analizie numerycznej

Definicje, twierdzenia

1 Stałopozycyjna i zmiennopozycyjna reprezentacja liczb

Liczby całkowite (typu **integer**). Dowolną liczbę całkowitą $l \neq 0$ możemy przedstawić w postaci skończonego rozwinięcia dwójkowego

$$l = s\sum_{i=0}^{n} e_i 2^i, \tag{1}$$

gdzie $s = \operatorname{sgn} l$, a $e_i \in \{0, 1\}$ $\quad (i = 0, 1, \ldots, n; \quad e_n = 1)$. W komputerze na reprezentację liczby przeznacza się słowo o skończonej długości np. $d+1$ bitów. Liczba l jest reprezentowalna w wybranej arytmetyce, jeśli tylko $n < d$. Dokładnie reprezentowane są liczby z przedziału $[-2^d + 1, 2^d - 1]$ (największa liczba dodatnia: $\underbrace{11\cdots 1}_{d \text{ razy}}$). Przy założeniu, że argumenty działania i jego wynik są reprezentowalne, dodawanie, odejmowanie i mnożenie liczb całkowitych (typu **integer**) jest wykonywane *dokładnie*. Zobaczymy, że nie jest tak, ogólnie biorąc, w wypadku liczb rzeczywistych (typu **real**).

Liczby rzeczywiste (typu **real**). Dowolną liczbę rzeczywistą $x \neq 0$ możemy przedstawić jednoznacznie w postaci

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

gdzie $s = \operatorname{sgn} x$, c jest liczbą całkowitą, zwaną *cechą* liczby x, a m jest liczbą rzeczywistą z przedziału $[\frac{1}{2}, 1)$, nazywaną *mantysą* tej liczby. d bitów słowa maszynowego ((d+1)-szy bit zawiera informację o znaku liczby) dzieli się na dwie części. Cechę c (wraz ze znakiem!) zapisuje się w sposób stałopozycyjny (por. (1)) na $d - t$ bitach słowa. Zakładamy, że c należy do przedziału liczb, które można przedstawić w ten sposób. Pozostałych t bitów słowa przeznacza się na reprezentację mantysy m. Ogólnie biorąc, zamiast nieskończonego rozwinięcia mantysy

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

zapamiętuje się cyfry *zaokrąglenia mantysy*, o skończonym rozwinięciu

$$m_t = \sum_{i=1}^{t} e^*_{-i} 2^{-i}, \quad \text{gdzie} \quad e^*_{-i} \in \{0, 1\}.$$

Zaokrąglenie symetryczne (ang. *symmetric rounding*):

$$m_t = \operatorname{rd}(m) := \sum_{i=1}^{t} e_{-i} 2^{-i} + e_{-(t+1)} 2^{-t}.$$

Twierdzenie 1.1 *Błąd zaokrąglenia mantysy nie przekracza* $\frac{1}{2}2^{-t}$*:*

$$| m - m_t | \leq \frac{1}{2} 2^{-t}.$$

Reprezentację liczby x stanowi zatem trójka (s, c, m_t), zapamiętywana w jednym słowie. Zero jest zwykle reprezentowane przez słowo złożone z samych zerowych bitów. Reprezentację zmiennopozycyjną liczby x będziemy oznaczać symbolem $\mathrm{rd}\,(x)$, gdzie

$$\mathrm{rd}\,(x) := s \cdot m_t^r \cdot 2^c.$$

Twierdzenie 1.2 (Błąd reprezentacji zmiennopozycyjnej liczby rzeczywistej) *Dla* $x \neq 0$ *zachodzi nierówność*

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

Inaczej mówiąc, zachodzi równość $\mathrm{rd}\,(x) = x(1+\varepsilon)$, *gdzie* ε *jest pewną liczbą o wartości bezwzględnej nieprzekraczającej* 2^{-t}.

Zbiór X ***liczb reprezentowalnych*** w arytmetyce zmiennopozycyjnej określamy następująco:

$$X := (-D, D), \quad \text{gdzie} \quad D := 2^{c_{\max}}, \qquad c_{\max} := 2^{d-t-1} - 1.$$

Niedomiar zmiennopozycyjny występuje wówczas, gdy $|x| < \frac{1}{2D}$.
Mamy do czynienia z ***nadmiarem zmiennopozycyjnym***, jeśli $|x| \geq D$.
Zbiór reprezentacji arytmetyki zmiennopozycyjnej definiujemy jako obraz zbioru X w odwzorowaniu rd, czyli $X_{fl} := \mathrm{rd}\,(X)$.
Zmiennopozycyjne działania arytmetyczne. Dla danych liczb zmiennopozycyjnych a, b i działania $\diamond \in \{+, -, \times, /\}$, zakładając, że $a \diamond b \in X_{fl}$ oraz $|a \diamond b| \geq \frac{1}{2D}$, definiujemy

$$fl(a \diamond b) := \mathrm{rd}\,(a \diamond b).$$

Błąd zmiennopozycyjnego działania arytmetycznego:

$$fl(a \diamond b) = (a \diamond b)(1 + \varepsilon_\diamond), \quad \text{gdzie} \quad \varepsilon_\diamond = \varepsilon_\diamond(a, b),\ |\varepsilon_\diamond| \leq 2^{-t}.$$

Definicja 1.3 *Mówimy, że liczba* $\tilde{x} \in X_{fl}$ *przybliża liczbę* $x \in X$ *z błędem względnym* ***na poziomie błędu (względnego) reprezentacji****, jeśli dla niewielkiej stałej* p *(np. rzędu jedności) zachodzi nierówność* $|\tilde{x} - x| \leq p2^{-t}|x|$.

Tak więc obliczony wynik działania arytmetycznego jest dokładnym wynikiem, zaburzonym na poziomie błędu reprezentacji. Następujące twierdzenie umożliwia upraszczanie wyrażeń dla oszacowań błędów, powstających w trakcie realizacji algorytmów obliczeniowych.

Twierdzenie 1.4 *Jeśli* $|\delta_i| \leq 2^{-t}$ *(*$i = 1, 2, \ldots, n$*), to zachodzi równość*

$$\prod_{i=1}^{n}(1 + \delta_i) = 1 + \sigma_n, \tag{2}$$

gdzie $\sigma_n \approx \sum_{i=1}^{n} \delta_i$. *Jeśli* $n2^{-t} < 2$, *to jest prawdziwe oszacowanie*

$$|\sigma_n| \leq \gamma_n, \ \textit{gdzie} \ \ \gamma_n := \frac{n2^{-t}}{1 - \frac{1}{2}n2^{-t}}. \tag{3}$$

Jeśli $n2^{-t} < 0.1$, *to*

$$|\sigma_n| \leq \gamma_n \leq n\ (1.06 \cdot 2^{-t}) = n2^{-t_1} \approx n2^{-t},$$

gdzie $t_1 := t - \log_2(1.06) = t - 0.08406$.

2 Utrata cyfr znaczących

Przykład 2.1 Rozważmy instrukcję podstawienia $y := x - \sin x$ i przypuśćmy, że w pewnym kroku obliczeń jest ona wykonywana dla $x = \frac{1}{30}$. Załóżmy, że komputer pracuje w dziesięciocyfrowej arytmetyce dziesiętnej. Otrzymamy wówczas

$$\begin{aligned} x &:= 0.3333333333 \times 10^{-1} \\ \sin x &:= 0.3332716084 \times 10^{-1} \\ x - \sin x &:= 0.0000617249 \times 10^{-1} \\ x - \sin x &:= 0.6172490000 \times 10^{-5} \end{aligned}$$

Dla porównania wynik dokładny: $x - \sin x := 0.6172496579716\ldots \times 10^{-5}$. Tak więc wynik obliczony ma o 4 cyfry dokładne mniej, niż dane!

Mamy tu do czynienia z ze zjawiskiem nazywanym utratą cyfr znaczących. Może być ono uważane za *piętę Achillesową* obliczeń zmiennopozycyjnych, i – wobec tego – powinno się go unikać za wszelką cenę!! Przy tym groźne są nie tylko rozległe zniszczenia wywołane przez pojedyncze działania (odejmowania, w gruncie rzeczy), lecz również powtarzające się wielokrotnie małe wstrząsy. W obu wypadkach wynik końcowy może być katastrofalny.

3 Normy wektorowe

Definicja 3.1 Normą wektorową *nazywamy nieujemną funkcję rzeczywistą* $\|\cdot\|$, *określoną w przestrzeni* $\mathbf{R}^n$, *o następujących własnościach (*$\mathbf{x}$, $\mathbf{y}$ *oznaczają dowolne wektory z* $\mathbf{R}^n$, α *-dowolną liczbę rzeczywistą):*

$$\begin{aligned} &\|\mathbf{x}\| > 0 \ \textit{dla}\ \mathbf{x} \neq 0; \\ &\|\alpha\mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|; \\ &\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|. \end{aligned}$$

Definicja 3.2 *Normy wektorowe Höldera wektora* $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbf{R}^n$ *definiujemy następującymi wzorami:*

$$\begin{aligned} \|\mathbf{x}\|_1 &:= \sum_{k=1}^{n} |x_k| \qquad (\textit{norma pierwsza}); \\ \|\mathbf{x}\|_2 &:= \left(\sum_{k=1}^{n} x_k^2\right)^{1/2} \qquad (\textit{norma euklidesowa}); \\ \|\mathbf{x}\|_\infty &:= \max_{1\leq k\leq n} |x_k| \qquad (\textit{norma maksymalna}). \end{aligned}$$

4 Reprezentacja *fl* wektorów

Przyjmując dla wektora $\mathbf{x} \in \mathbf{R}^n$ reprezentację

$$\mathrm{rd}\,(\mathbf{x}) := (\mathrm{rd}\,(x_1), \ldots, \mathrm{rd}\,(x_n))^T,$$

otrzymujemy:

$$\|\mathbf{x} - \mathrm{rd}\,(\mathbf{x})\|_2 \leq 2^{-t}\|\mathbf{x}\|_2,$$

lub, w równoważnej postaci,

$$\mathrm{rd}\,(\mathbf{x}) = \mathrm{diag}\,(1 + \epsilon_i)\mathbf{x} \qquad (|\epsilon_i| \leq 2^{-t}),$$

gdzie $\mathrm{diag}\,(c_i) \in \mathbf{R}^{n\times n}$ oznacza macierz przekątniową, o elementach $c_1, \ldots, c_n$ na przekątnej głównej.

Takie samo oszacowanie zachodzi dla innych norm Höldera.

5 Uwarunkowanie zadania

Definicja 5.1 *Jeśli niewielkie względne zmiany danych zadania powodują duże względne zmiany jego rozwiązania, to zadanie takie nazywamy* ***źle uwarunkowanym****. Wielkości charakteryzujące wpływ zaburzeń danych na odkształcenia rozwiązania nazywamy* ***wskaźnikami uwarunkowania*** *zadania.*

Przykład 5.2 W wypadku zadania obliczenia wartości funkcji f w punkcie x, jeśli x zostanie lekko zaburzone o wielkość h, a więc jeśli $|h/x|$ będzie względnym zaburzeniem x, to

$$\frac{|f(x+h)-f(x)|}{|f(x)|} \approx \frac{|hf'(x)|}{|f(x)|} = \frac{|xf'(x)|}{|f(x)|}\frac{|h|}{|x|}.$$

Tak więc czynnik $C(x) := |xf'(x)|/|f(x)|$ można traktować jako ***wskaźnik uwarunkowania*** dla tego zadania. Jeśli $C(x)$ jest małe, to względna zmiana wyniku również będzie mała — zadanie jest dobrze uwarunkowane. Jeśli $C(x)$ jest duże, mała zmiana argumentu x może wywołać (bardzo) duże względne odkształcenie wyniku — wówczas mamy do czynienia z zadaniem źle uwarunkowanym!

6 Algorytmy numerycznie poprawne

Ogólnie biorąc, przy numerycznym rozwiązywaniu zadania w miejsce dokładnych danych pojawią się dane zaburzone przez błędy reprezentacji. Z drugiej strony, nawet jeśli znamy dokładne rozwiązanie dla tych danych, to w arytmetyce fl jest ono reprezentowane tylko w sposób przybliżony. Z tych względów

za numerycznie najwyższej jakości uznamy takie algorytmy, dla których obliczone rozwiązanie jest mało zaburzonym rozwiązaniem dokładnym dla mało zaburzonych danych.
Algorytmy o powyższej własności nazywamy ***numerycznie poprawnymi.***
Wariant sytuacji, gdy obliczone rozwiązanie jest rozwiązaniem dokładnym (a więc niezaburzonym) dla mało zaburzonych danych, wiąże się z pojęciem ***algorytmu numerycznie bardzo poprawnego.***

Przez „małe zaburzenia" rozumiemy tu zaburzenia na poziomie błędu reprezentacji.

Przykład 6.1 Rozważmy zadanie sumowania w naturalnej kolejności n liczb $x_1, x_2, \ldots, x_n$. Można wykazać, że

$$fl(\cdots((x_1+x_2)+x_3)+\cdots+x_n) = \tilde{x}_1+\tilde{x}_2+\cdots+\tilde{x}_n,$$

gdzie

$$\begin{aligned}
\tilde{x}_i &= x_i(1+\delta_i) \quad (i=1,2,\ldots,n),\\
1+\delta_i &:= \prod_{j=i}^{n}(1+\epsilon_j) \quad (i=1,2,\ldots,n),\\
\epsilon_1 &:= 0; \qquad |\epsilon_i| \le 2^{-t} \qquad (i=2,3,\ldots,n).
\end{aligned}$$

Na mocy tw. 1.4

$$|\delta_1| \le \gamma_{n-1}, \qquad |\delta_i| \le \gamma_{n+1-i} \quad (i=2,3,\ldots,n).$$

Oznacza to, że obliczony wynik jest równy ***dokładnej sumie*** nieco zaburzonych danych — algorytm sumowania jest więc numerycznie bardzo poprawny.