Ćwiczenia z Metod Numerycznych I A
rok akademicki 2001/2001
seria 1

Tomasz Masłowski Jakub Narębski Tomasz Pawłowski
Przemysław Olbratowski

26 października 2001 roku

1 Dokładność numeryczna

Liczby rzeczywiste są przechowywane w komputerze ze skończoną dokładnością. Skończony jest także zakres tych liczb. W związku z tym przy obliczeniach na liczbach rzeczywistych pojawiają się błędy numeryczne. Musimy zatem z różnych poprawnych matematycznie algorytmów wybrać taki, który powoduje najmniejszy błąd związany z prowadzeniem obliczeń na komputerze.

Przy pomocy prostych przykładów chcielibyśmy zilustrować, że nie każdy poprawny matematycznie algorytm obliczania danego wzoru jest poprawny numerycznie. Często użycie złego algorytmu prowadzi do znacznego błędu.

1.1 Reprezentacja liczb rzeczywistych (zmiennoprzecinkowych)

Każdą liczbę x, $x \neq 0$ można reprezentować jednoznacznie w postaci wykładniczej $x = s \cdot m \cdot 2^c$, gdzie $s \in \{-1, 1\}$ jest znakiem liczby, $m \in [\frac{1}{2}, 1)$ nazywamy **mantysą**, a $c \in \mathbb{Z}$ nazywamy **cechą**. W komputerze m i c są reprezentowane za pomocą skończonej ilości bitów. W związku ze skończonym zakresem wartości przyjmowanych przez cechę $c_{\min} \leq c \leq c_{\max}$ w reprezentacji zmiennoprzecinkowej można reprezentować tylko liczby ze skończonego zakresu $\frac{1}{2}2^{c_{\min}} \leq |x| < 2^{c_{\max}}$ oraz 0. W związku ze skończoną ilością t bitów mantysy reprezentacja zmiennoprzecinkowa może różnić się od wartości ścisłej: $\tilde{x}_t = rd(x) = x(1+\epsilon)$, gdzie $|\epsilon| \leq 2^{-t}$.

Możliwe źródła błędów numerycznych

- Błędy danych wejściowych, czy to związane z pomiarem, czy to powstałe w wyniku obcięcia przy przejściu do reprezentacji zmiennoprzecinkowej.
- Błędy obcięcia powstałe przez zastąpienie granicy (np. sumy nieskończonej) przez n–ty wyraz ciągu (sumę częściową).
- Błędy zaokrągleń wynikające z prowadzenia obliczeń na liczbach zmiennoprzecinkowych:

$$fl(a \Box b) = (a \Box b)(1 + \epsilon(a, b, \Box))$$

Przykładowo przy dodawaniu (odejmowaniu) dwu liczb zmiennoprzecinkowych najpierw sprowadza się je przeskalowywując do wspólnej cechy, dodaje mantysy a następnie odrzuca bity poza zakresem.

Zadanie 1. Znaleźć najmniejsze takie $a \neq 0$, że w obliczeniach zmiennoprzecinkowych $fl(1+a)=1$. Znaleźć taką liczbę a, że $fl(1/a*a) \neq 1$. *Uwaga:* w razie potrzeby zastosować zmienne pomocnicze (volatile) aby uniknąć optymalizacji kompilatora.

1.2 Unikanie nadmiaru i niedomiaru (w obliczeniach pośrednich)

Chcemy obliczyć wartość bezwzględną liczby zespolonej $z = a + ib \neq 0$ lub długość przeciwprostokątnej trójkąta o przyprostokątnych a i b, czy też odległość euklidesową między dwoma punktami na płaszczyźnie.

Klasyczny algorytm $|z| = \sqrt{a^2+b^2}$ może prowadzić do nadmiaru lub niedomiaru w obliczeniach pośrenich, podczas gdy wynik jest reprezentowalny w arytmetyce zmiennoprzecinkowej.

Lepszy algorytm wygląda następująco

if $(|a| > |b|)$ **then**

$\quad \mathrm{z} = |a|\sqrt{1+\left(\frac{|b|}{|a|}\right)^2}$

else

$\quad \mathrm{z} = |b|\sqrt{1+\left(\frac{|a|}{|b|}\right)^2}$

Zadanie 2. Zaimplementować oba te algorytmy. Porównać ich dokładność oraz dokładność funkcji bibliotecznej `hypot` (z `math.h`). Dla jakich a i b metoda klasyczna nie działa, a druga metoda daje poprawne wyniki (do tego nie trzeba użyć komputera)?

1.3 Redukcja cyfr znaczących przy odejmowaniu bliskich sobie liczb

1.3.1 Szacowanie błędu algorytmu na przykładzie wzoru $a^2 - b^2$

Istnieją dwa algorytmy obliczania wzoru $a^2 - b^2$.

$$\begin{aligned} &a^2 - b^2 \\ &(a-b)(a+b) \end{aligned} \tag{1}$$

Każde działanie numeryczne generuje pewien błąd w porównaniu do rzeczywistej wartości wyniku dzialania:

$$\begin{aligned} fl(a+b) &= (a+b)(1+\epsilon) \\ fl(a*b) &= (a*b)(1+\epsilon) \\ fl(\sqrt{a}) &= \sqrt{s}(1+\epsilon), \end{aligned} \tag{2}$$

gdzie poprzez $fl(a \Box b)$ oznaczamy wynik numerycznego dzialania $\Box$.

Oszacujmy błędy otrzymane w obu algorytmach:

$$\begin{aligned} fl(a^2-b^2) &= (a*a(1+\epsilon_1)-b*b(1+\epsilon_2))(1+\epsilon_3) \\ &= (a^2-b^2)\left(1+\frac{a^2\epsilon_1-b^2\epsilon_2}{a^2-b^2}\right)(1+\epsilon_3) \\ &= (a^2-b^2)(1+\delta), \end{aligned} \tag{3}$$

gdzie $\delta = \frac{a^2\epsilon_1-b^2\epsilon_2}{a^2-b^2}(1-\epsilon_3)+\epsilon_3$. Jeżeli a^2 jest odpowiednio bliskie b^2, a ϵ_1 i ϵ_2 maja przeciwne znaki, to błąd względny δ wyniku otrzymanego pierwszym algorytmem może być dowolnie duży.

Nie jest to prawda w przypadku drugiego algorytmu:

$$\begin{aligned} fl((a-b)(a+b)) &= ((a-b)(1+\epsilon_1)(a+b)(1+\epsilon_2))(1+\epsilon_3) \\ &= (a^2-b^2)(1+\delta), \end{aligned} \tag{4}$$

gdzie $\delta \leq |\epsilon_1|+|\epsilon_2|+|\epsilon_3|$.

1.3.2 Obliczanie zer równania kwadratowego

$$x^2-2px+q=0 \tag{5}$$

W szkole stosuje się algorytm:

$$x_1* = p+\sqrt{p\cdot p-q} \tag{6a}$$
$$x_2 = p-\sqrt{p\cdot p-q} \tag{6b}$$

Poprawny algorytm wygląda następująco: obliczamy ten pierwiastek dla którego mamy we wzorze dodawanie, a drugi obliczamy na podstawie wzoru Verleta $x_1\cdot x_2=q$. Zapis tego algorytmu poniżej

`if` $p \geq 0$ `then`

$$\begin{aligned} x_1 &= p+\sqrt{p\cdot p-q} \\ x_2 &= q/x_1 \end{aligned} \tag{7}$$

`if` $p < 0$ `then`

$$\begin{aligned} x_2 &= p-\sqrt{p\cdot p-q} \\ x_1 &= q/x_2 \end{aligned} \tag{8}$$

Kiedy algorytm 'szkolny' jest niepoprawny? Może on generować znaczny błąd w chwili gdy np. $p\cdot p \gg q$.

Zadanie 3. Zaimplementować oba te algorytmy. Poszukać takich wartości p i q, dla których algorytmy dają różne wyniki. Który algorytm jest dokładniejszy?

Przygotował: Jakub Narębski