Metody Numeryczne
Zajęcia nr 1

Michał Bernardelli

Do zapamiętania: arytmetyka zmiennoprzecinkowa, uwarunkowanie zadania, stabilność numeryczna, poprawność numeryczna, złożoność obliczeniowa

1 Arytmetyka zmiennoprzecinkowa

Liczby rzeczywiste są przechowywane w komputerze ze skończoną dokładnością i w ograniczonym tylko zakresie. Wpływa to na różnice między wynikami poprawnymi a tymi obliczonymi na maszynie cyfrowej. Nie każdy zatem poprawny matematycznie algorytm jest także poprawny numerycznie. Użycie nieodpowiedniego algorytmu może prowadzić do znaczących *błędów numerycznych*.

W odróżnieniu od arytmetyki matematycznej – idealnej, na komputerach stosuje się **arytmetykę zmiennoprzecinkową**. Inne spotykane jej nazwy to *arytmetyka numeryczna* czy *arytmetyka fl* (od ang. *floating-point arithmetic*).

Każda liczba rzeczywista $x \neq 0$ może być w jednoznaczny sposób przedstawiona w postaci

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

gdzie:

- β jest bazą (podstawą) systemu liczbowego, najczęściej równą 2 lub 10,
- $s \in \{-1, 1\}$ jest znakiem liczby x,
- $c \in \mathbb{Z}$ nazywamy **cechą** lub *wykładnikiem*,
- $m \in \left[\frac{1}{\beta}, 1\right)$ nazywamy **mantysą** lub *ułamkiem*.

Podkreślmy wyraźnie, że pierwsza cyfra mantysy jest *niezerowa*. W ogólnym przypadku zarówno mantysa jak i cecha, mogą mieć nieskończenie wiele cyfr w rozwinięciu, ale w arytmetyce fl reprezentowane są za pomocą skończonej liczby bitów. Z rozwinięcia mantysy pamiętanych jest t cyfr znaczących, a ostatnia cyfra jest zaokrąglana. Cecha c może przyjmować wartości z zakresu $(c_{\min}, c_{\max})$. W związku z tym w arytmetyce fl (o podstawie dziesięć) można reprezentować tylko liczby z przedziału $(0.1 \cdot 10^{c_{\min}}, 10^{c_{\max}})$. Wartość $\widetilde{x}$ w arytmetyce fl może się zatem różnić się od wartości faktycznej x:

$$\widetilde{x} = rd(x) = x(1+\varepsilon), \qquad \text{gdzie } |\varepsilon| \leq \nu = 5 \cdot 10^{-t}.$$

Na współczesnych komputerach liczby reprezentowane są w standardzie IEEE-754 (*Institute of Electrical and Electronics Engineers, 1985*). Niedokładności wynikające ze stosowania arytmetyki fl:

- *błędy reprezentacji* – niedokładności przy reprezentacji liczb (danych),

- *błędy zaokrągleń* – niedokładności związane z wykonywaniem działań arytmetycznych. Zakładamy, że działania arytmetyczne wykonywane na reprezentacjach liczb rzeczywistych $\widetilde{x}$ i $\widetilde{y}$ są wykonywane dokładnie, a wynik jest przedstawiany w arytmetyce fl, czyli

  $$fl_\nu(x \square y) = rd(\, rd(x) \square rd(y)\,) = [x(1+\varepsilon_x) \square y(1+\varepsilon_y)]\,(1+\varepsilon_\square),$$

  gdzie $|\varepsilon_x| \leq \nu$, $|\varepsilon_y| \leq \nu$ i $|\varepsilon_\square| \leq K_\square \cdot \nu$ dla pewnej stałej $K_\square$.

Przykład 1
W systemie dwójkowym liczba $\frac{1}{10}$ ma rozwinięcie nieskończone:

$$(0.1)_{10} \approx (0.000110011001100110011001100110)_2$$

2 Poprawność numeryczna

Niech $\widetilde{x}$ będzie przybliżeniem dokładnej wartości x. Zdefiniujmy
Błąd bezwzględny jako $\Delta x = |\widetilde{x} - x|$.
Błąd względny dla $x \neq 0$ jako $\left|\frac{\Delta x}{x}\right|$.
Błędy względne czasami wyraża się procentowo.
Zadania, w których niewielkie względne zaburzenia danych powodują niewielkie względne zmiany jego rozwiązania nazywamy **dobrze uwarunkowanymi**. Wielkości charakteryzujące wpływ zaburzeń danych na zaburzenia rozwiązania zadania nazywamy **wskaźnikami uwarunkowania** zadania.

Przykład 2
Klasycznym przykładem zadania, które jest źle uwarunkowane, jest wyznaczenie pierwiastków wielomianu

$$w(x) = (x-1)(x-2)\cdot\ldots\cdot(x-20) = x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + \ldots + 2432902008176640000.$$

Oczywiście pierwiastkami są liczby rzeczywiste: $1, 2, \ldots, 20$*. Jeżeli zmienimy współczynnik przy* x^{20} *na* $1,0000001$*, czyli zaburzymy go o* 10^{-7}*, to wielomian będzie miał teraz* 14 *pierwiastków zespolonych i 6 rzeczywistych. Zatem niewielkie względne zaburzenie współczynników (danych zadania) powoduje duże względne zmiany pierwiastków (rozwiązań zadania).*

Algorytm rozwiązywania zadania nazywamy **numerycznie poprawnym** jeśli dla każdej danej wynik działania algorytmu można w arytmetyce zmiennoprzecinkowej zinterpretować jako nieco zaburzony wynik działania algorytmu w arytmetyce idealnej dla nieco zaburzonej informacji na temat danych, przy czym poziom zaburzeń nie zależy od danych.
Podkreślmy, że numeryczna poprawność jest cechą algorytmu, zaś dobre uwarunkowanie cechą zadania. Oznacza to, że nawet dobry, poprawny numerycznie algorytm może dawać nieprawidłowe wyniki dla zadań źle uwarunkowanych.
Istotną kwestią jest także **złożoność obliczeniowa** zadania, czyli istnienie i własności optymalnych metod rozwiązania zadania oraz ich konstrukcji. Podstawowym kryterium jest liczba działań arytmetycznych potrzebnych do rozwiązania zadania.
Następną cechą algorytmu jest jego **stabilność**. Algorytm numeryczny nie jest stabilny, jeżeli małe błędy popełnione na pewnym etapie jego realizacji mają wpływ na znaczący spadek dokładności następnych obliczeń.

Przykład 3 (algorytm Hornera)
Jednym z najczęściej spotykanych zadań obliczeniowych jest wyznaczenie wartości wielomianu

$$w(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

w punkcie x. Zadanie to można rozwiązać przynajmniej na dwa sposoby, różniące się znacznie kosztem (i co ważniejsze poprawnością numeryczną).

Sposób I *(2n mnożeń)*

$w := a_0$
$p := 1$
dla $i = 1, 2, \ldots, n$
    $p := p * x$
    $w := w + a_i * p$

Sposób II *(n mnożeń)*

$$w(x) = (\ldots((a_n x + a_{n-1})x + a_{n-2})x + \ldots + a_1)x + a_0$$

$w := a_n$
dla $i = n-1, n-2, \ldots, 0$
    $w := w * x + a_i$

Zadanie 1 (jak małe jest zero)
Zaproponować sposób na sprawdzenie dokładności reprezentacji liczb rzeczywistych. Znaleźć w Microsoft Excel najmniejsze takie m, n, *dla których* $10^{-m} = 0$ *oraz* $2^{-n} = 0$.

Zadanie 2 (residuum układu równań liniowych)
Dla zadanego układu równań

$$\begin{pmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.8642 \\ 0.1440 \end{pmatrix}$$

oraz przybliżonego rozwiązania

$$x = [0.9911, -0.4870]^T$$

wyznaczyć residuum układu, to jest wielkość

$$r = b - Ax.$$

oraz normę drugą residuum. Norma druga wektora $y \in \mathbb{R}^n$ *dana jest wzorem:*

$$\|y\|_2 = \sqrt{\sum_{i=1}^{n} y_i^2}.$$

Czy na podstawie tych wielkości można coś wnioskować na temat dokładności rozwiązania?
Wyznaczyć residuum oraz jego normę drugą dla prawdziwego rozwiązania:

$$x = [2, -2]^T.$$

Wskazówka: *w programie Microsoft Excel użyć formuł* MACIERZ.ODW *oraz* MACIERZ.ILOCZYN. *Pamiętać, że są to formuły tablicowe i do ich zatwierdzenia należy użyć kombinacji klawiszy CTRL+SHIFT+ENTER.*

Zadanie 3 (odwracanie macierzy Hilberta)
Macierz Hilberta dana jest wzorem (dla $n \geq 1$)

$$H_n = \begin{pmatrix} \frac{1}{1+1-1} & \frac{1}{1+2-1} & \cdots & \frac{1}{1+n-1} \\ \frac{1}{2+1-1} & \frac{1}{2+2-1} & \cdots & \frac{1}{2+n-1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n+1-1} & \frac{1}{n+2-1} & \cdots & \frac{1}{n+n-1} \end{pmatrix}$$

Dla $n \in \{3, 5\}$ obliczyć macierz odwrotną do H_n, a następnie sprawdzić wyliczenia poprzez obliczenie macierzy $H_n H_n^{-1}$ i porównanie jej z macierzą jednostkową.

Zadanie 4 (układ równań z macierzą Hilberta)
W programie Microsoft Excel rozwiązać układ równań z macierzą Hilberta H_n oraz wektora prawej strony złożonego z samych jedynek. Zadanie wykonać dla $n \in \{5, 10, 20\}$. Sprawdzić uzyskane wyniki poprzez obliczenie normy drugiej residuum.

Zadanie 5 (długość przeciwprostokątnej)
Zaproponować poprawny numerycznie algorytm obliczania długości przeciwprostokątnej trójkąta c o przyprostokątnych a i b. Klasyczny algorytm $c = \sqrt{a^2 + b^2}$ może prowadzić do nadmiaru lub niedomiaru w obliczeniach pośrednich. Dla jakich a i b metoda klasyczna nie działa?

Zadanie 6
Zapisać algorytm rozwiązywania układu równań $Dx = b$ z macierzą diagonalną D. Uwzględnić przypadki gdy macierz jest osobliwa. Jaka jest złożoność obliczeniowa zaproponowanego algorytmu?

Zadanie 7
Zapisać w pseudokodzie algorytm rozwiązywania układu równań z nieosobliwą macierzą dolną trójkątną L i wektorem prawej strony b. Jaki jest koszt działania algorytmu?

Zadanie 8 (obliczanie exp)
Zaproponować algorytm obliczania wartości funkcji e^x oparty na wzorze Taylora:

$$e^x = \sum_{k=0}^{n} \frac{x^k}{k!} + \underbrace{\frac{1}{(n+1)!} e^{\xi} x^{n+1}}_{R_n(x)},$$

gdzie ξ jest punktem pośrednim pomiędzy 0 a x. Jako kryterium stopu przyjąć ograniczenie na liczbę iteracji oraz wielkość poprawki w kolejnym k-tym kroku iteracji, to jest $\left|\frac{x^k}{k!}\right| < \delta$.
Napisać program w Octave i sprawdzić jego działanie dla $x \in \{100, 25, 1, 0, -1, -25, -100\}$. Porównać wyniki z wynikami zwracanymi przez funkcję exp(). *Czy program daje poprawne wyniki dla wszystkich $x \in \mathbb{R}$?*
(*) Ocenić a priori, ile składników należy obliczyć w celu osiągnięcia zadanej dokładności $\varepsilon > 0$.