Ćwiczenia z ANALIZY NUMERYCZNEJ (M)

Blok 1: lista M 3

18 października 2017 r.

M3.1. $\boxed{\text{1,5 punktu}}$ Załóżmy, że x, y są liczbami maszynowymi, tzn. $\mathrm{rd}\,(x) = x, \mathrm{rd}\,(y) = y$, takimi, że $0 < y < x$. Wykazać, że jeśli

$$2^{-q} \leqslant 1 - \frac{y}{x} \leqslant 2^{-p}$$

(p i q są całkowite), to

$$p \leqslant \text{liczba bitów straconych przy odejmowaniu } x - y \leqslant q.$$

M3.2. $\boxed{\text{1 punkt}}$ Wartość wielomianu $L(x) := a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$ w punkcie x można obliczyć według następującego *schematu Hornera*:

— Oblicz wielkości pomocnicze $w_0, w_1, \ldots, w_n$ za pomocą wzorów
  a) $w_n := a_n$,
  b) $w_k := w_{k+1} \times x + a_k \quad (k = n-1, n-2, \ldots, 0)$.
— Wynik: $L(x) = w_0$.

Zakładając, że $a_0, a_1, \ldots, a_n$ oraz x są liczbami zmiennopozycyjnymi wykazać, że schemat Hornera jest algorytmem numerycznie poprawnym.

M3.3. $\boxed{\text{1 punkt}}$ Wartość sumy $\sum_{k=1}^{n} a_k$, gdzie $n := 2^m$ dla pewnego $m \in \mathbb{N}$, można wyznaczyć stosując strategię *dziel i zwyciężaj*. Np. dla $m = 3$ obliczenia wykonywane są wówczas zgodnie z następującym diagramem:

$$\begin{array}{ccccccccccccccc}
a_1 & & a_2 & & a_3 & & a_4 & & a_5 & & a_6 & & a_7 & & a_8 \\
\searrow & & \swarrow & & \searrow & & \swarrow & & \searrow & & \swarrow & & \searrow & & \swarrow \\
& s_{12} & & & & s_{34} & & & & s_{56} & & & & s_{78} & \\
& \searrow & & & & \swarrow & & & & \searrow & & & & \swarrow & \\
& & & s_{14} & & & & & & & & s_{58} & & & \\
& & & \searrow & & & & & & & & \swarrow & & & \\
& & & & & & & s_{18} & & & & & & &
\end{array}$$

gdzie $s_{ij} := a_i + a_{i+1} + \ldots + a_j$. Wykazać, że ten algorytm jest numerycznie poprawny i — dla dużych wartości n — dokładniejszy (na ogół) niż zwykły algorytm sumowania.

M3.4. $\boxed{\text{1 punkt}}$ Pole n-kąta foremnego ($n \geqslant 4$) wpisanego w okrąg o promieniu 1 wynosi

$$P_n = \frac{1}{2} n \sin \frac{2\pi}{n}.$$

Wartość P_n jest przybliżeniem liczby π – tym lepszym, im większe jest n. Następujący algorytm pozwala oszczędnie obliczać kolejno P_4, P_8, $P_{16}, \ldots$:

$$s_2 := 1, \quad c_2 := 0, \quad P_4 := 2;$$
$$s_k := \sqrt{\tfrac{1}{2}(1 - c_{k-1})}, \quad c_k := \sqrt{\tfrac{1}{2}(1 + c_{k-1})}, \quad P_{2^k} := 2^{k-1}\, s_k \qquad (k = 3, 4, \ldots).$$

a) Uzasadnić powyższy algorytm.

b) Stosując wybraną arytmetykę t-cyfrową ($t \geqslant 128$) obliczyć P_{2^k} dla $k = 2, 3, \ldots, 2t$.

c) Czy wyniki są zgodne z oczekiwaniami? Jeśli nie, to jakie jest źródło kłopotów? Jak można ich uniknąć?

M3.5. $\boxed{\text{1,5 punktu}}$ Załóżmy, że $f'(x) > 0$ i $f''(x) > 0$ dla $x \in \mathbb{R}$. Ponadto, niech α będzie pierwiastkiem równania $f(x) = 0$. Wykazać, że jest to jedyny pierwiastek, a metoda Newtona daje ciąg do niego zbieżny dla dowolnego przybliżenia początkowego x_0.

M3.6. $\boxed{\text{1,5 punktu}}$ Uzasadnić, że odwrotność liczby c można obliczać bez wykonywania dzieleń, za pomocą wzoru $x_{n+1} := x_n(2 - c\,x_n) \quad (n = 0, 1, \ldots)$. Uzasadnić (lokalną?) zbieżność tej metody. Dla jakich wartości x_0 metoda jest zbieżna?

M3.7. $\boxed{\text{1 punkt}}$ Podać przykład funkcji $f \in C^2[a, b]$ oraz przybliżenia początkowego $x_0 \in [a, b]$, dla którego ciąg przybliżeń otrzymany za pomocą metody Newtona jest zbieżny liniowio do pierwiastka funkcji f.

M3.8. $\boxed{\text{1 punkt}}$ Zaprogramować w języku Julia metodę Steffensena

$$c_{n+1} = c_n - \frac{[f(c_n)]^2}{f[c_n + f(c_n)] - f(c_n)},$$

a następnie zastosować ją do znalezienia pierwiastka równania

$$e^{-x} - \sin x = 0.$$

Zastosować arytmetykę wysokiej precyzji (np. $t \geqslant 128$), aby móc podać eksperymentalną wartość wykładnika zbieżności (lokalnej?) tej metody.

10 października 2017

Rafał Nowak