Ćwiczenia
z ANALIZY NUMERYCZNEJ (L)

Lista nr 1

7 października 2020 r.

Zajęcia 13 października 2020 r.
Zaliczenie listy **od 5 pkt.**

L1.1. **Włącz komputer!** | 1 punkt | Niech dana będzie funkcja $f(x) = 4040\dfrac{\sqrt{x^{11}+1}-1}{x^{11}}$. Napisz program, który działając w trybie podwójnej precyzji (`double`) obliczy wartość $f(0.001)$. Czy wynik jest wiarygodny? **Odpowiedź uzasadnij**.

L1.2. **Włącz komputer!** | 1 punkt | Niech dana będzie funkcja $f(x) := 12120\dfrac{x-\sin x}{x^3}$. Przy pomocy komputera oblicz w arytmetyce pojedynczej (`single`) i podwójnej precyzji (`double`) wartości $f(10^{-i})$ dla $i = 11, 12, \ldots, 20$. Czy otrzymane wyniki są poprawne? **Odpowiedź uzasadnij**.

L1.3. **Włącz komputer!** | 1 punkt | Liczby rzeczywiste $y_0, y_1, \ldots$ są zdefiniowane rekurencyjnie w następujący sposób:

$$y_0 = 1, \qquad y_1 = -\frac{1}{7}, \qquad y_{n+2} = \frac{1}{7}(69y_{n+1} + 10y_n) \qquad (n = 0, 1, \ldots).$$

Użyj komputera i podanej zależności do obliczenia (w pojedynczej lub podwójnej precyzji) kolejno wartości liczb $y_2, y_3, \ldots, y_{50}$. Skomentuj otrzymane wyniki. Czy są one wiarygodne? **Odpowiedź uzasadnij**.

L1.4. **Włącz komputer!** | 2 punkt | Sprawdź, że całki

$$I_n := \int_0^1 \frac{x^n}{x+2020}\mathrm{d}x \qquad (n = 0, 1, \ldots)$$

spełniają następującą zależność rekurencyjną:

$$I_n = \frac{1}{n} - 2020I_{n-1} \qquad \left(n = 1, 2, \ldots;\ I_0 = \ln\frac{2021}{2020}\right). \tag{1}$$

Następnie wykorzystaj związek (1) do wyznaczenia wartości całek $I_1, I_2, \ldots, I_{20}$ (w takiej właśnie kolejności) wykonując obliczenia w arytmetyce pojedynczej lub podójnej precyzji używając pętli `for`. Czy wyniki są wiarygodne? **Odpowiedź uzasadnij**.

L1.5. **Włącz komputer!** | 1 punkt | Wykorzystując własności szeregów naprzemiennych, ustal ilu teoretycznie wyrazów szeregu

$$\pi = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}$$

należy użyć do obliczenia wartości π z błędem mniejszym niż 10^{-4}. Następnie wykonaj odpowiedni eksperyment obliczeniowy przy pomocy komputera w arytmetyce pojedynczej lub podwójnej precyzji. Co z niego wynika?

L1.6. $\boxed{\text{1 punkt}}$ Wykorzystując własności szeregów naprzemiennych, sprawdź, że do obliczenia wartości $\ln 2$ z błędem mniejszym niż $\frac{1}{2} \cdot 10^{-6}$ trzeba użyć ok. dwóch milionów wyrazów szeregu

$$\ln x = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(x-1)^k}{k}$$

dla $x = 2$. Wykaż, że zastosowanie prostego związku $\ln 2 = \ln[e(2/e)]$ może znacznie przyspieszyć obliczenia.

L1.7. $\boxed{\text{1 punkt}}$ W języku programowania `PWO++`[1] funkcja `ATG`(x) oblicza z bardzo dużą dokładnością wartość $\operatorname{arctg}(x)$, jednak **tylko wtedy**, gdy $|x| \leq 1$. Wykorzystując funkcję `ATG`, zaproponuj szkic `algorytmu` wyznaczającego w języku `PWO++` wartości funkcji arcus tangens z dużą dokładnością także dla $|x| > 1$.

L1.8. $\boxed{\textbf{Włącz komputer!}}\boxed{\text{2 punkty}}$ Na wykładzie pokazano, że użycie wzoru

$$\frac{f(x+h) - f(x)}{h} \qquad (h \text{ -- małe})$$

do przybliżenia wartości $f'(x)$ nie jest dobrym pomysłem. Uzasadnij, że

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h},$$

a następnie zbadaj doświadczalnie **dla wielu** doborów f oraz x przydatność wyrażenia

$$\frac{f(x+h) - f(x-h)}{2h} \qquad (h \text{ -- małe})$$

do wyznaczania przybliżonej wartości pochodnej funkcji f w punkcie x. Czy stosowanie drugiego wzoru coś zmienia? Jak to wytłumaczyć?

(–) Paweł Woźny

– Czyli, że zasadniczo Pan się musi na tym rozeznać całkowicie żeby wiedzieć ile i gdzie...

– Dotychczas tak było, ale teraz mamy komputer. Może Pan pisać co tylko Pan chce to nie ma żadnego znaczenia.

– Komputer?

– Eeee, on się i tak zawsze pomyli przy dodawaniu, proszę pana. Nie było miesiąca, żeby się nie pomylił.

– Czyli, że teraz nie trzeba się tak znać na robocie?

– A teraz już nie. Teraz jest dużo łatwiej, jest proszę pana.

– Komputer...

Miś, reż. S. Bareja, 1980 (1:24:26).

P.S. Film można obejrzeć przed ćwiczeniami, ale nie jest to konieczne do zaliczenia listy.

[1] Jak powszechnie wiadomo od wielu lat, jest to najlepszy język programowania. Dlatego rekomendujemy jego używanie nie tylko na analizie numerycznej, ale i na co dzień ;-)