Algorytmy numeryczne – Zadanie 1

Arkadiusz Dąbrowski
grupa T1
nr indeksu 246859

14.10.2018

Sumowanie szeregów potęgowych

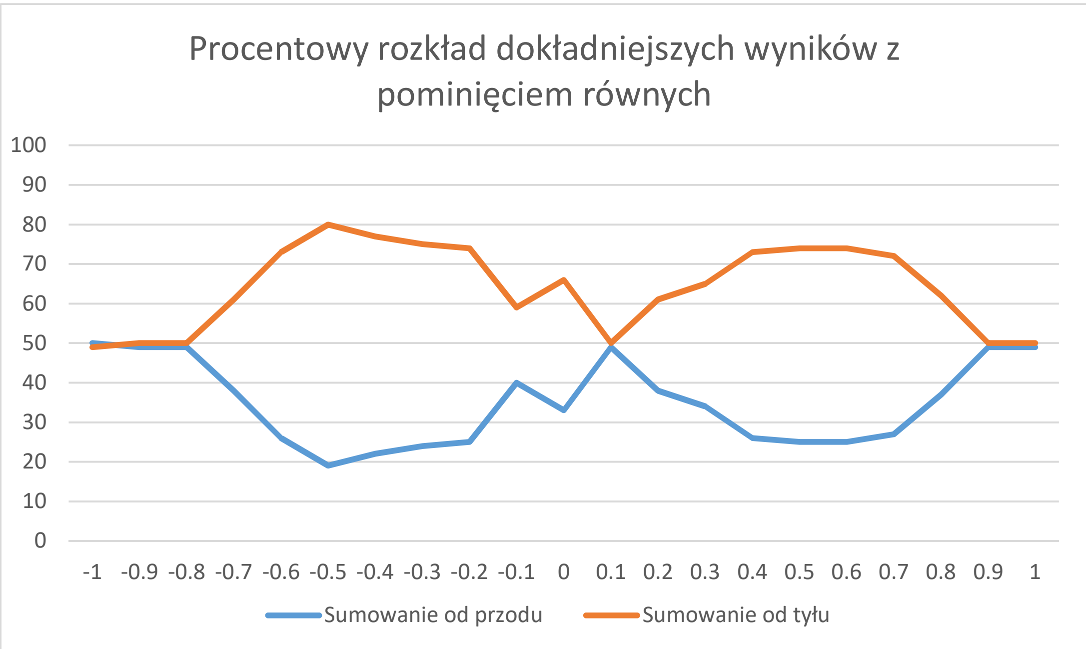
Poniższe sprawozdanie prezentuje próby wyliczenia funkcji $\ln(1+x)\cdot \operatorname{arctg}(x)$ za pomocą szeregu potęgowego wykorzystując do tego implementację w języku Java. Próby te zostały wykonane na zmiennej typu double na cztery różne sposoby:

1. sumując elementy szeregu potęgowego obliczane bezpośrednio ze wzoru Taylora
$$\ln(1+x) = \sum_{n=1}^{\infty}(-1)^{n+1}\frac{x^n}{n} \; oraz \; \operatorname{arctg}(x) = \sum_{n=0}^{\infty}(-1)^n\frac{x^{2n+1}}{2n+1}$$
   a. w kolejności od początku,
   b. w kolejności od końca,
2. sumując elementy szeregu potęgowego obliczając kolejny wyraz na podstawie poprzedniego
$$dla \; \ln(1+x)\colon a_n = \frac{-x\cdot(n-1)}{n} \; oraz \; dla \; \operatorname{arctg}(x)\colon \; a_{n+1} = a\cdot -\frac{x^2\cdot(2n+1)}{2n+3}$$
   a. w kolejności od początku,
   b. w kolejności od końca.

Dane szeregi są zbieżne w przedziale $x \in (-1, 1]$ dlatego zakres badanych argumentów został ograniczony do tego przedziału. Obliczenia zostały wykonane dla $1e^6$ (miliona) argumentów, gdzie dla każdego z nich zsumowano 150 kolejnych składników szeregu, co w sumie daje $1.5e^8$ różnych próbek. Otrzymane wyniki zostały porównane z bibliotecznymi funkcjami wbudowanymi, co pozwoliło na zweryfikowanie poniższych hipotez.

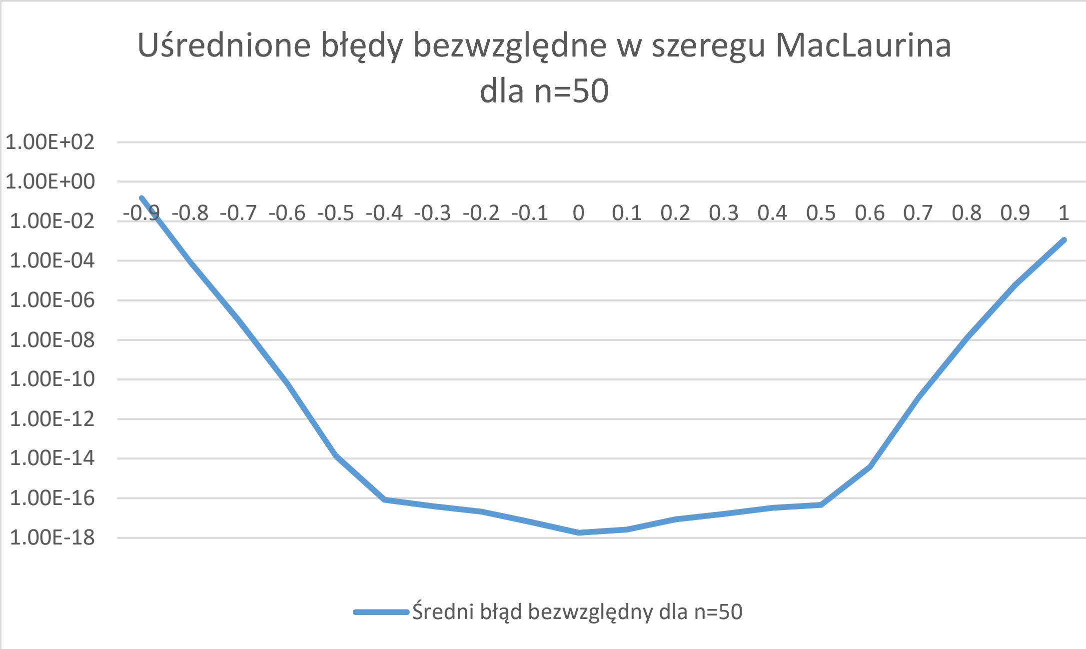
Hipotezy

H1: sumowanie od końca daje dokładniejsze wyniki niż sumowanie od początku.



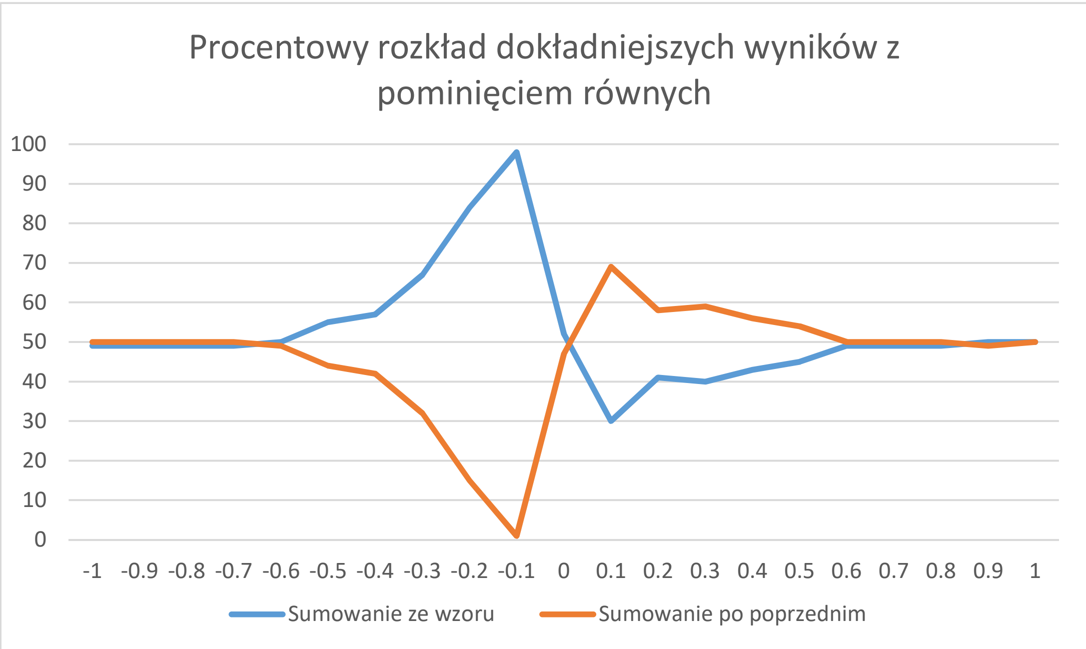
Z wykresu obok wynika, że zależnie od argumentu dokładniejsze wyniki dostajemy średnio dla 66% wprowadzanych argumentów, gdzie najmniejsza różnica jest na krańcach przedziału oraz w okolicach 0, a największa w okolicach -0.5 oraz 0.5, gdzie aż w 80% przypadków sumowanie od tyłu było dokładniejsze, co potwierdza postawioną hipotezę.

H2: używającrozwinięcia wokół 0 (szereg MacLaurina), przy tej samej liczbie składników szeregu dokładniejsze wyniki uzyskujemy przy małych argumentach.



Aby to sprawdzić wyliczone zostały średnie błędów bezwględnych na małych przedziałach, gdzie liczba składników n została przyjęta jako liczba stała równa 50. Na wykresie widać że w okolicy zera, gdzie argumenty są najmniejsze średni błąd bezwzględny jest również najmniejszy, co jednoznacznie potwierdza postawioną hipotezę

H3: sumowanie elementów obliczanych na podstawie poprzedniego daje dokładniejsze wyniki niż obliczanych bezpośrednio ze wzoru.

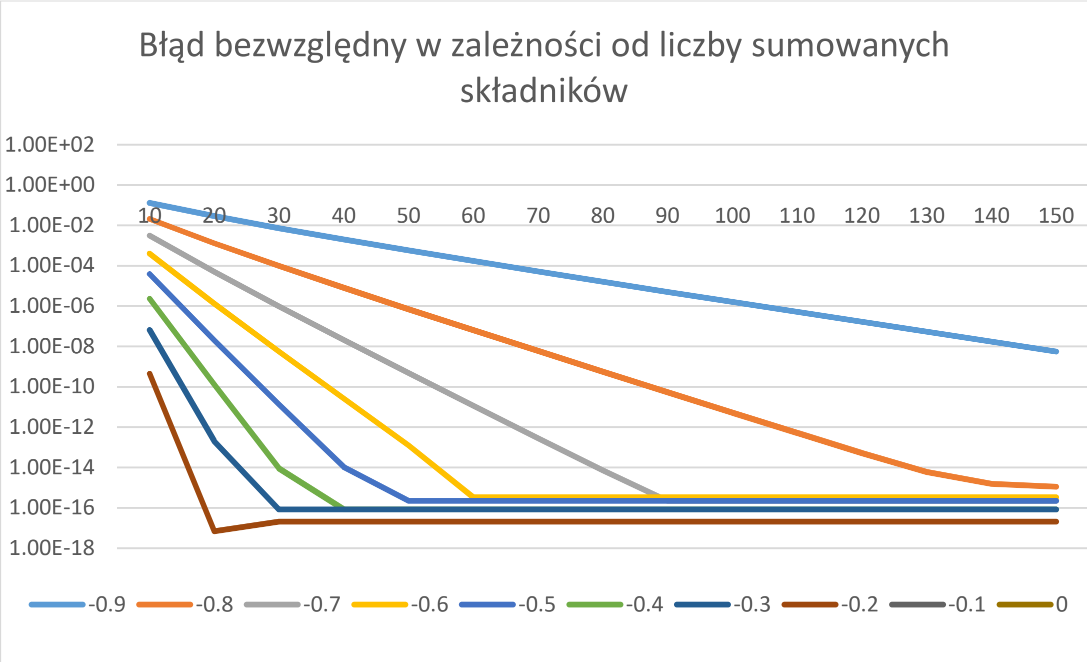


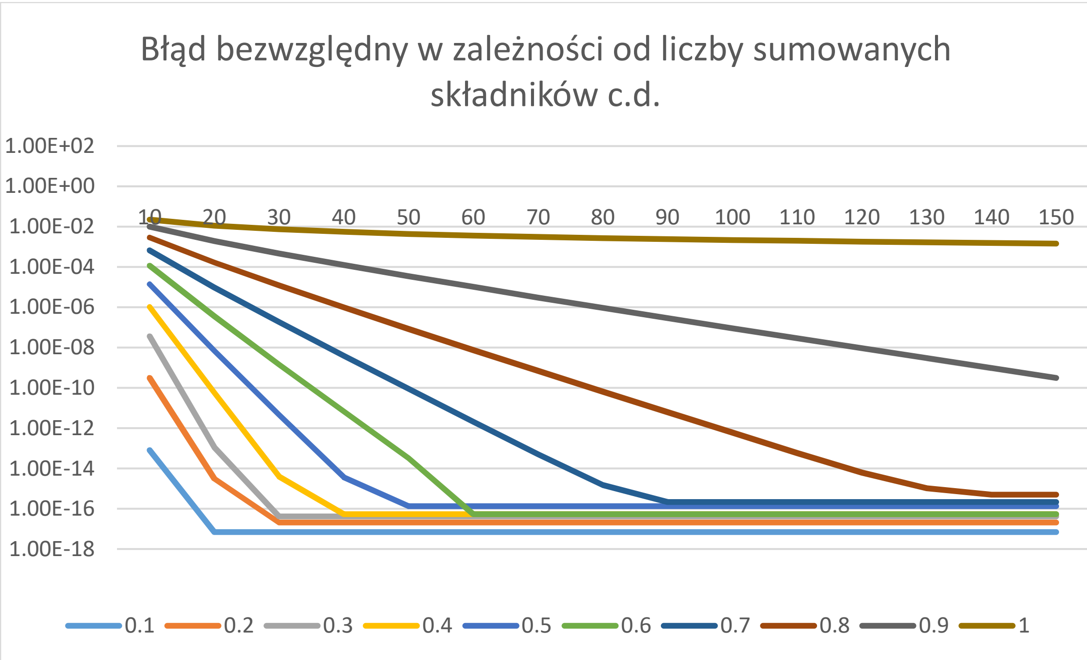
Z danego wykresu wynika, że zależnie dla danego przedziału różnica między sumowaniem ze wzoru a sumowaniem po porzednim wyrazie jest na zbliżonym poziomie $\sim 50\%$. Jednak sumowanie na podstawie poprzedniego elementu jest zdecydowanie dokładniejsze dla przedziału $(0, 0.6)$.

Pytania

Q1: Jak zależy dokładność obliczeń (błąd) od liczby sumowanych składników?

Dokładność obliczeń rośnie liniowo dla coraz większej liczby sumowanych składników, dla wszystkich argumentów zadanej funkcji, co możemy zaobserwować na poniższych wykresach. Dodatkowo można zauważyć, że dla każdego n od pewnego momentu błąd bezwzględny jest stały, co bezpośrednio wynika ze specyfiki zmiennoprzecinkowego typu double, który odrzuca pewne bardzo małe wartości, aby pomieścić najważniejsze wartości.


Błąd bezwzględny w zależności od liczby sumowanych składników
1.00E+02
1.00E+00
1.00E-02
1.00E-04
1.00E-06
1.00E-08
1.00E-10
1.00E-12
1.00E-14
1.00E-16
1.00E-18
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
-0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0


Błąd bezwzględny w zależności od liczby sumowanych składników c.d.
1.00E+02
1.00E+00
1.00E-02
1.00E-04
1.00E-06
1.00E-08
1.00E-10
1.00E-12
1.00E-14
1.00E-16
1.00E-18
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1