Jakub Balcerowicz
Indeks: 238153
Grupa 1 Tester-Programista
Zadanie 1

Sumowanie szeregów potęgowych

Do obliczania wartosci funckji arcus tangens posłuży nam program napisany w języku C. Wartości funkcji obliczamy na dwa sposoby, ze wzoru ogólnego:

$$arctg(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}$$

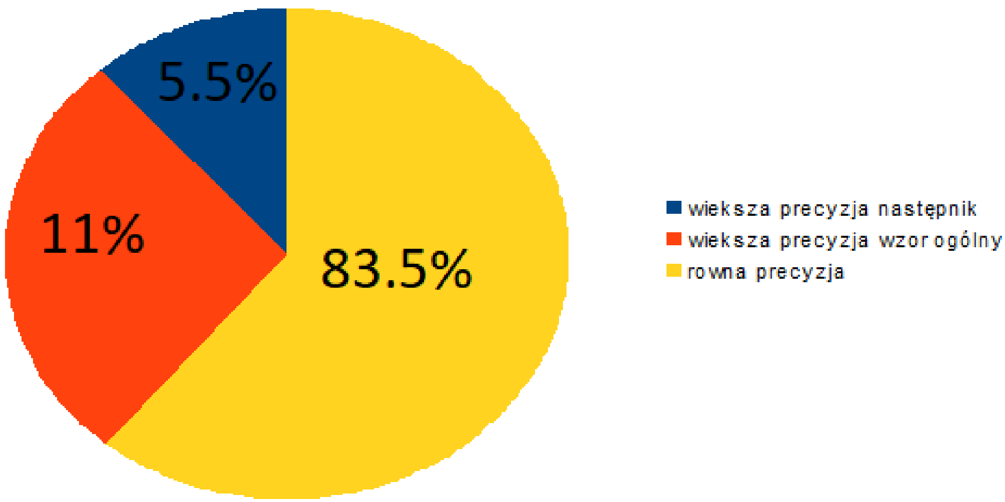
oraz obliczając kolejny wyraz szeregu na podstawie poprzedniego:

$$a_{n+1} = a * -\frac{x^2 * (2n+1)}{2n+3}$$

W obu przypadkach mamy do czynienia z sumą szeregu. Dla każdego sposobu będziemy sumować szereg od początku i od końca. Obliczenia wykonamy na zmiennej double w przedziale od [-1,1] (przedział zbieżności szeregu).

W pierwszym kroku porównujemy w obu sposobach sumowania od początku i od końca. Przeprowadzamy porównanie na przedziale [-1,1]. W przedziałach [-1,-0.9] i [0.9,1] sumowanie od tyłu jest precyzjniejsze w ok.93%. W przedziale [-0.9,0.9] precyzja sumowania od tyłu wynosi ok.98.5%.

W następnym kroku badamy błąd bezwzględny obu algorytmów sumujących od tyłu. Punktem odniesienia jest funkcja z biblioteki C atan(). Otrzymane wyniki porównujemy tj. błąd wzoru ogólnego z błędem ze sposobu z poprzednikiem. Wykres pokazuje, że oba algorytmy mają podobny współczynik precyzji.



Tak duża dysproporcja między sumowaniem od początku i od końca wynika z tego,że gdy do coraz większej liczby dodajemy liczbę coraz mniejszą to w pewnym momencie dojdziemy do tego,że A + a = A. Dlatego sumowanie od tyłu jest lepsze co pokazuje krok pierwszy, w którym ok 98% jest bardziej precyzyjne.

Podsumowując, sumowanie od początku nie jest precyzyjne. Mój algorytm wskazuje na to, że najbradziej precyzyjnym sposobem na liczenie wartości arctan jest sposób ze wzorem ogólnym. Czynniki mające wpływ na zaistniałe błędy to zbyt mała ilość zsumowanych elementów w szeregu, używanie operacji dzielenia przy której jest możliwoć utraty danych.