Algorytmy numeryczne/Projekt nr 1
Sumowanie szeregów potęgowych
25 – 10 – 2017 r.

Dawid Więcko
Numer indeksu: 238154
Spec.: tester programista
Grupa: II
Typ liczbowy: double

Moim zadaniem było zbadanie sumowania szeregu potęgowego **arctg(z)**.
Zadanie zostało wykonane na 5 sposobów:
- korzystając z funkcji bibliotecznej
- sumowanie elementów szeregu od początku do końca
- sumowanie elementów szeregu od końca do początku
- obliczając kolejny wyraz ciągu na podstawie poprzedniego oraz bezpośrednio ze wzoru od początku do końca
- obliczając kolejny wyraz ciągu na podstawie poprzedniego oraz bezpośrednio ze wzoru od końca do początku

Zakres jaki badałem to liczby od -0.99 do 0.99. Dla wartości powyżej 2 pojawia się wartość nan (błąd precyzji poprzez nadmiar – overflow). Powodem jest to, że dla innych wartości należałoby rozpatrywać wiele przypadków (wzorów) z powodu zbieżność I rozbieżność szeregu potęgowego. Można zauważyć znaczący skok wartości na wartościach granicznych (-1 oraz 1). Z tego powodu odrzuciłem je, aby wynik był jak najbardziej wiarygodny.

Suma szeregu potęgowego

$$\arctan(z) = z - \frac{z^3}{3} + \frac{z^5}{5} - \frac{z^7}{7} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n+1}}{2n+1}\,; \qquad |z| \le 1 \qquad z \ne i, -i$$

Suma szeregu na podstawie poprzednika

$$\lim_{n \to +\infty} \left| \frac{u_{n+1}}{u_n} \right|.$$

$$\frac{u_{n+1}}{u_n} = (-1)^{n+1} \cdot \frac{x^{2n+2}}{2n+3} \frac{2n+1}{(-1)^n x^{2n}} = -\frac{2n+1}{2n+3} x^2.$$

Średnia wartość bezwzględna dla 4 metod

Próba nr 1

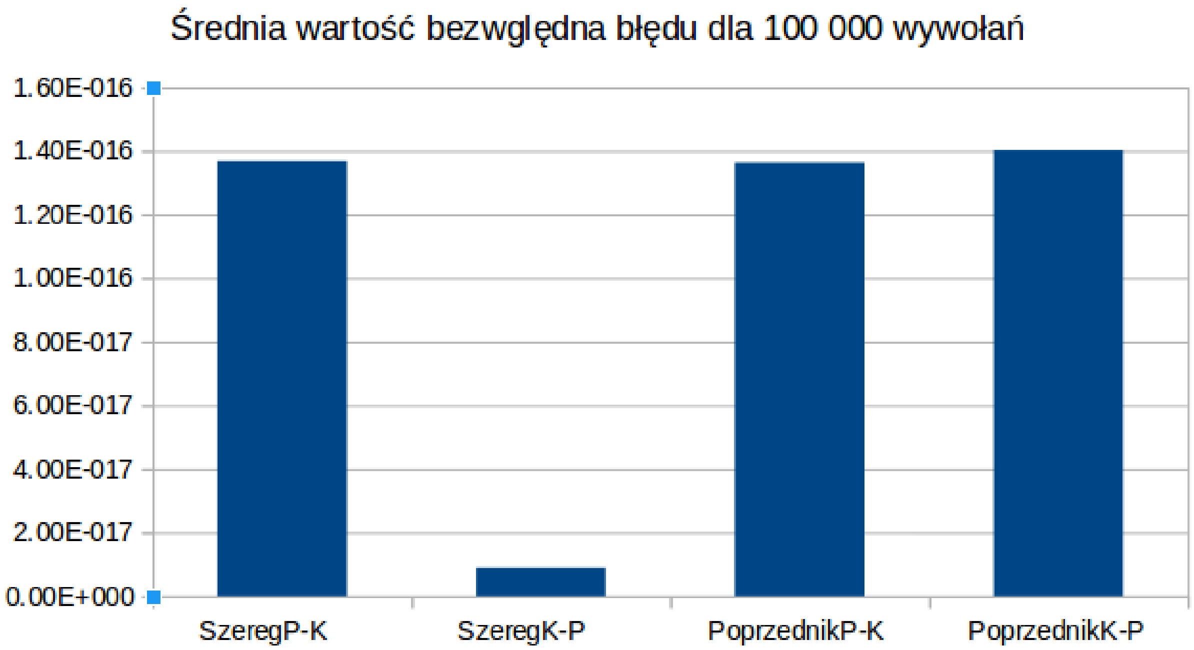
N: 10
SzeregP-K 0.000965925159634008234123270231208380209864117205 14
SzeregK-P 0.00096592515963400747518174949135527640464715659618
PoprzednikP-K 0.00065097310164949611505091908725262328516691923141
PoprzednikK-P 0.00065097310164949503084874660174818927771411836147
Zwycięzca: **PoprzednikK-P**

Próba nr 2

N: 100
SzeregP-K 0.0000016717301302676558963475512975538350701754097 8
SzeregK-P 0.0000016717301301957137882472177389803213998220599 0
PoprzednikP-K 0.0000015438695891668998596814527460296950778229074 8
PoprzednikK-P 0.0000015438695891679931674581212341923475150906597 3
Zwycięzca: **PoprzednikP-K**

Próba nr 3

Przedstawiam tylko najistotniejszą próbę, ponieważ jej wartości są jak najbardziej dokładne I istotne dla całego zadania.



N: 100 000
SzeregP-K 0.00000000000000013677528000480417549990474181826736
SzeregK-P 0.00000000000000000898551715637279403953082869217283
PoprzednikP-K 0.00000000000000013634072301284230125174177500810814
PoprzednikK-P 0.00000000000000014023771797301645919605528157579904
Zwycięzca: **SzeregK-P**

Badania przeprowadziłem na przedziale od <-0.99,0.99> ze skokiem 0.001. Przedstawiają one bardzo ciekawą zależność dotyczącą ilości wywołań.

W przypadku **10** wywołań pętli błąd był największy dla metody szeregu od początku do końca, a najlepszą metodą okazała się być metoda poprzednika od końca do początku.

W przypadku **100** wywołań najgorsza metoda to szeregu od początku do końca, a najlepszą metoda to sumowanie poprzednika od początku do końca.

W przypadku **100 000** wywołań najlepszą metodą I najdokładniejszą tym razem okazała się być metoda sumowania szeregu taylora od końca do początku.

Trend ten jest także wzrostowy dla tej metody. Testowałem wyniki także dla 500 000 wywołań i wyniki w tym przypadku były takie same.

Próbowałem testować dla **100 000 000**
Jednak tutaj liczenie pojedynczej wartości zajmowało mojemu procesorowi około 30s przy 100% obciążeniu, dlatego musiałem zmniejszyć skok do 0.1. Serwer studencki Sigma jak I kompilatory online odmawiały współpracy “wieszając” się, bądź zwracając komunikat o przekroczeniu czasu. Do wartości 0.40 cały czas metoda od końca do początku ze wzoru na sumę szeregu potęgowego wskazywała cały czas wartość 0.

N 100 000 000
Skok 0.1
SzeregP-K 0.00000000000000014571677198205179593060165643692017
SzeregK-P 0.00000000000000000277555756156289150513347472120592
PoprzednikP-K 0.00000000000000014016565685892600829810284046514228
PoprzednikK-P 0.00000000000000014294121442048891444030389252934068

Jak widać na powyższym przykładzie im wyższa liczba wywołań tym dokładniejszy wynik.

Nie mniej jednak rezultaty tego eksperymentu są dość ciekawe.
Podając za N (ilość wywołań) bardzo małą liczbę ilość kroków jest znikoma. Przy błędach obcięcia jest jednak bardzo istotne, ponieważ im więcej dodajemy do siebie podobnych wartości tym większa szansa na lepszy wynik. Dodając do siebie wiele liczb różnica pomiędzy nimi jest podobna I tym samym precyzja. Dzięki temu możemy uniknąć błędów obcięcia, które pojawiają się, gdy dodamy zbyt dużą liczbę do małej. Zwiększenie wykładnika o 1 podwaja zakres, ale liczby ją reprezentujące są o połowę mniejsze. Przez tą operację mniejsza liczba traci swoją precyzję.