

Kordian Ceranowski 253897

Sprawozdanie z zadania 1 z przedmiotu Algorytmy Numeryczne – wariant Ln

Zadanie wykonałem w języku Java

Zadanie rozpocząłem od zbudowania klasy Ln zawierającej w sobie 2 obiekty typu ArrayList w których trzymane są kolejne wyliczone wyrazy ciągu.

Następnie zaimplementowałem funkcje:

<i>fillArrayNaive()</i>	-	wypełniającą tablicę wyliczając kolejne wyrazy ciągu Taylora
<i>fillArraySmart()</i>	-	wypełniającą tablicę wyliczając kolejne wyrazy mnożąc wcześniejsze
<i>getSum()</i>	-	zwracającą sumę wyrazów dodawanych w normalnej kolejności
<i>getReversedSum()</i>	-	zwracającą sumę wyrazów dodawanych w odwrotnej kolejności
<i>getRelativeError</i>	-	zwracającą błąd względny otrzymanego wyniku
<i>precisionTest</i>	-	sprawdza ile podejść potrzeba do uzyskania dokładności 10^{-6}

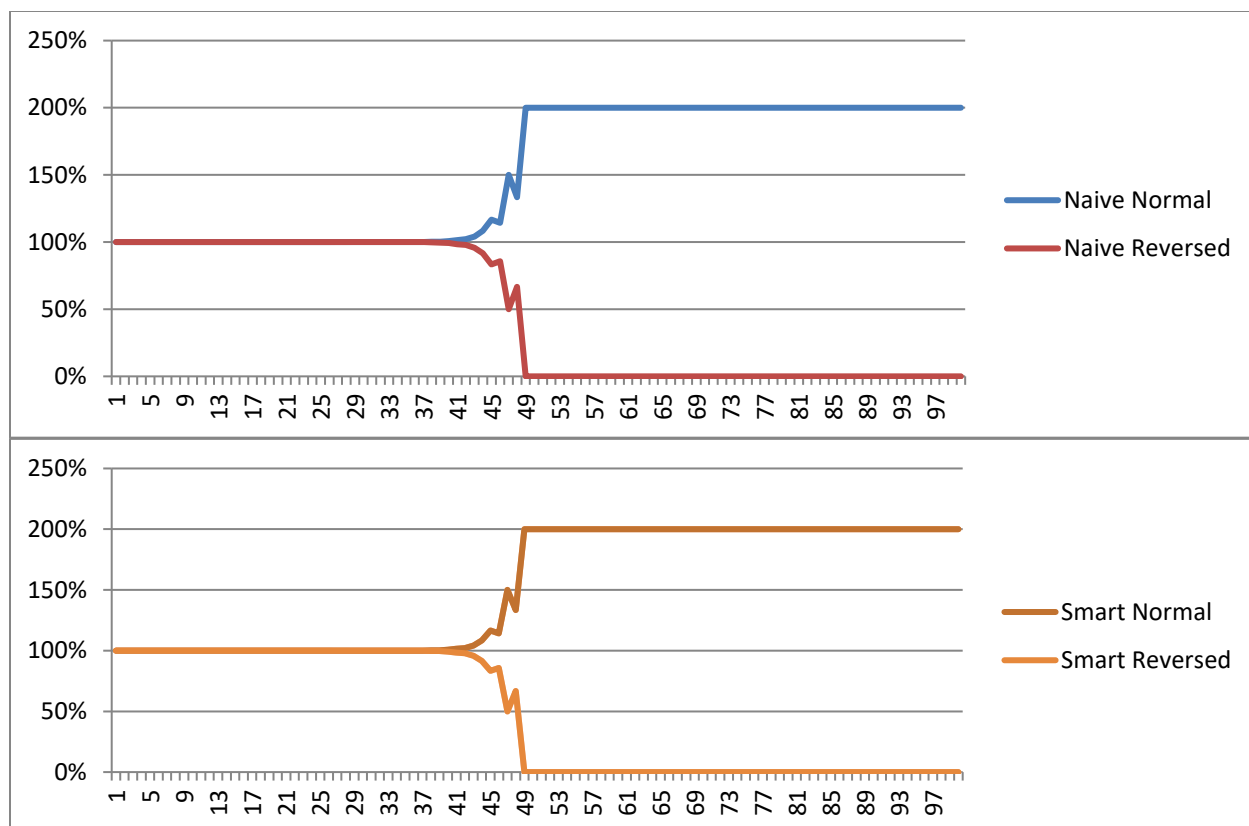
Następnie utworzyłem klasę GraphDataStorage wraz z klasą wewnętrzną GraphIndex

Zawarłem w niej tablicę trzymającą wartości błędów bezwzględnych w 4 pomiarach dla danego N z metodami umożliwiającymi wypełnienie jej rekordami z podanego zakresu N.

Zawarłem tam również funkcję packToCSV() umożliwiającą przeniesienie wszystkich wyników do pliku CSV.

Jako ostatnią zbudowałem również klasę AverageErrorDataStorage umożliwiającą przyrównanie do siebie wyników błędów, co umożliwiło mi stwierdzenie który ze sposobów umożliwia uzyskanie nadokładniejszych danych. Metody w niej zawarte tworzą plik CSV z danymi odnośnie bliskości wyniku danej metody pomiarowej względem średniego wyniku wszystkich metod.

Aby nie używać metody wbudowanej Math.pow stworzyłem również interfejs MyMath w którym umieściłem prostą metodę pow wyliczającą potęgę $var1^{var2}$.



Wykres dla $\ln(0.5)$

Linia pozioma oznacza ilość prób, a więc dokładność. 100% oznacza, że dany pomiar dał błąd tej samej wielkości co średnia błąd z wszystkich pomiarów.

Z powyższych wykresów wynika, że metoda mnożenia poprzednich wyrazów nie przyniosła żadnych zmian w dokładności wyników, jednak sumowanie składników od końca, tj. Od najmniejszego do największego okazało się być metodą bardzo zwiększającą dokładność pomiaru.

Po przeprowadzeniu dodatkowych pomiarów stwierdziłem również, że algorytm osiąga maksymalną dokładność możliwą do otrzymania w typie double już po średnio 60, a w krytycznych przypadkach przy około 200 wyliczonych wyrazach.

Po wykonaniu testu na 100000 rekordów nie stwierdziłem żadnych zmian w wynikach

HIPOTEZY

H1. Prawda

H2. Fałsz, nie zauważono różnicy

H3. Fałsz, nie zauważono różnicy

Q1. Dokładność obliczeń rośnie razem z liczbą sumowanych składników

Q2. Dodatkowa funkcja *precisionTest()* wskazuje, że dla małych wartości x wymagana ilość składników potrafi bardzo wzrosnąć, natomiast dla większych staje się relatywnie mała
Prezentuje to zjawisko poniższy wykres

