26.10.2017

Algorytmy numeryczne
Zadanie 1: Sumowanie szeregów potęgowych

Jan Bienias
nr indeksu 238201
Informatyka III (Tester programista), grupa 1

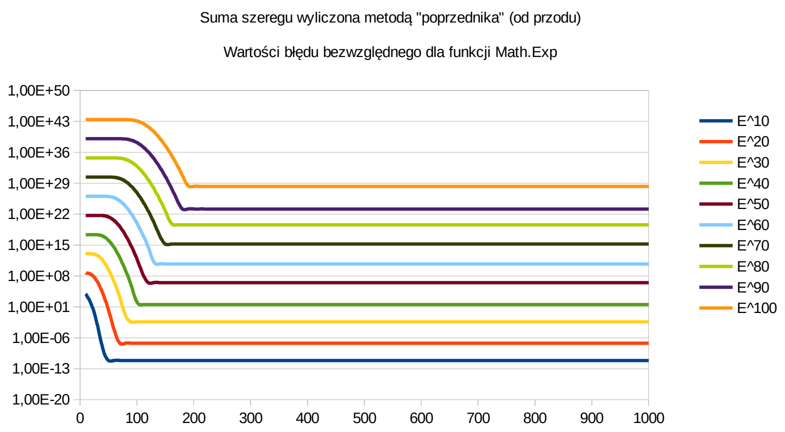
1. Wstęp
Celem zadania jest zaimplementowanie rozwiązania pozwalającego na zsumowanie szeregu potęgowego e^x przy pomocy funkcji wbudowanej danego (wybranego) języka programowania, znalezienie i określenie dokładności (precyzji) owej funkcji danego (wybranego) języka programowania względem własnej implementacji arytmetycznego sumowania rozwinięcia szeregu z użyciem wzoru na nast. element, grupując elementy od przodu i od tyłu oraz odpowiedź na pytanie, jakie parametry musi przyjąć dana funkcja, by dzięki niej uzyskać możliwie najdokładniejszy względem funkcji wbudowanej wynik.

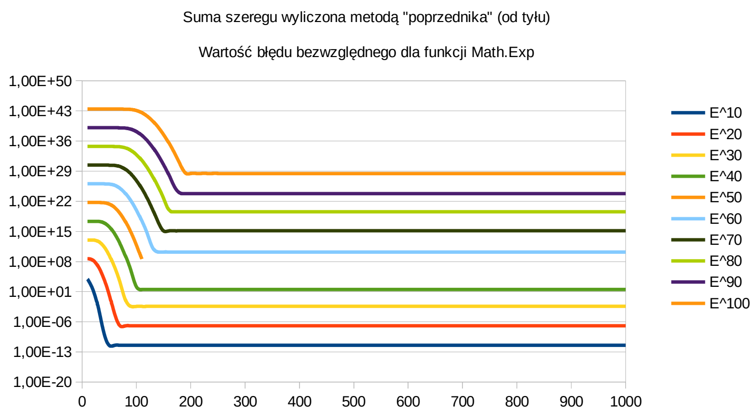
2. Materiały i metody
Dokładne obliczenia zostały dokonane:
- z użyciem funkcji bibliotecznej `Math.Exp` języka C#
- z użyciem własnej implementacji funkcji liczącej kolejne sumy wyrazów szeregu $S_{n+1}=\frac{x}{n+1}*S_n$
- z użyciem własnej implementacji funkcji liczącej kolejne sumy wyrazów szeregu $e^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}$

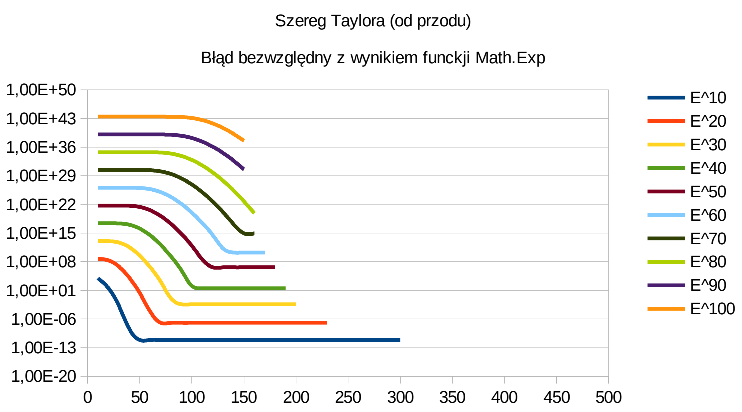
Zostały dodatkowo podzielone na:
- sumowanie elementów kolejno od przodu,
- sumowanie elementów kolejno od tyłu.

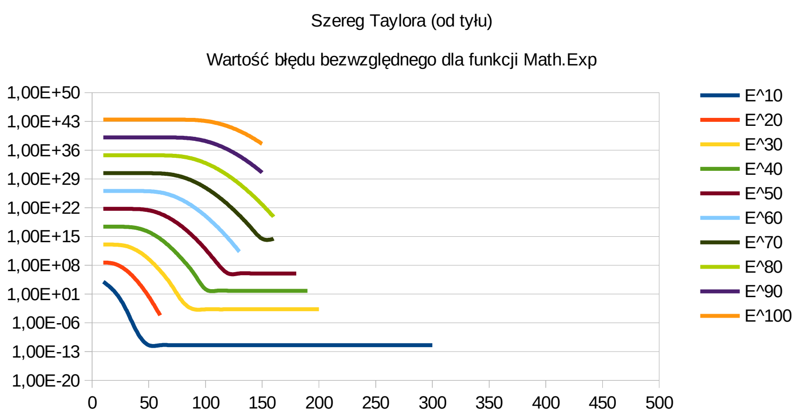
Wszystkie powyższe wartości liczono na zmiennych typu `double`.

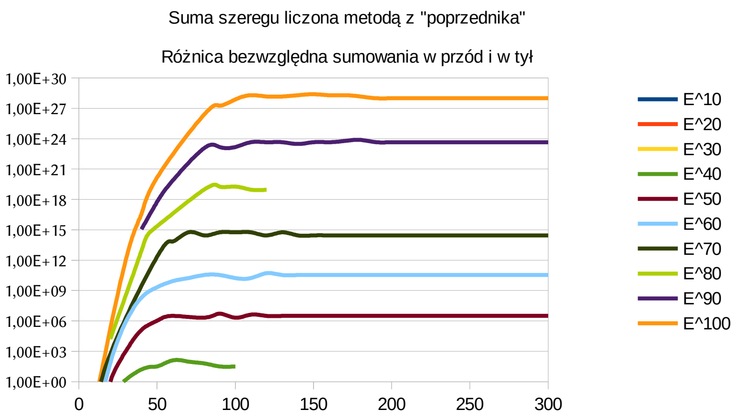
3. Wyniki

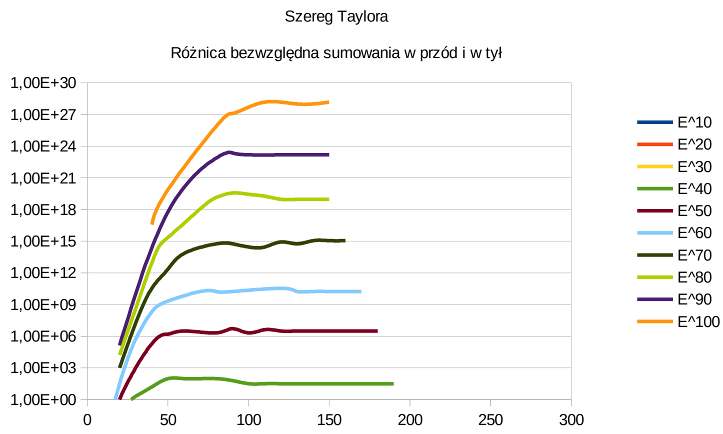












Powyższe wykresy przedstawiają obliczenia wykonane dla x w przedziale [10, 100], inkrementowane o 10 oraz n w przedziale [10, 1000], inkrementowane o 10. Puste miejsca na powyższych wykresach oznaczają osiągnięcie wartości niereprezentowalnych **`NaN`** (lub w przypadku e^{50} na wykresie dot. sumy szeregu wyliczonej metodą „poprzednika” od tyłu – **0**).

4. Wnioski

- Ze względu na wykonywanie obliczeń w typie `double` możemy uzyskać przybliżenie do ok. 15 cyfry po przecinku. Ma to bezpośredni wpływ na precyzję wyniku sumowania.
- Sumowanie od tyłu jest dokładniejsze niż sumowanie od przodu, ponieważ dodajemy liczby mniejsze do coraz to większych. Zmienna `double` odrzuca końcowe cyfry mniejszej liczby aby pomieścić najistotniejsze wartości.
- Wartości błędów bezwzględnych między funkcją wbudowaną a funkcjami zaimplementowanymi powtarzają się od pewnego n, co oznacza, że osiągnęliśmy maksymalne możliwe przybliżenie.
- Różnice bezwzględne między metodami liczenia „w przód” i „w tył” obydwu zaimplementowanych funkcji przyjmują bardzo zbliżone wartości.
- Próba implementacji funkcji liczącej kolejne sumy wyrazów szeregu (ze wzoru Taylora), przy użyciu zmiennych typu `double`, nie przyniosła oczekiwanych rezultatów – wartości liczone tą metodą przekraczały zakres już przy $n=[140,300]$, ponieważ funkcja rośnie wykładniczo.