Gdańsk, 13 października 2018

Algorytmy Numeryczne – Zadanie 1

Sumowanie szeregów potęgowych

Andrzej Chorostian
Informatyka 1st. 3r. gr.T1

1. Temat

Napisać program obliczający wartości zadanej funkcji: $e^{\sin(x)}$ na 4 sposoby:
Sumując elementy szeregu potęgowego obliczane bezpośrednio ze wzoru Taylora i na podstawie poprzedniego elementu. Oba te sposoby należy wykonać w wersjach z sumowaniem elementów w kolejności od początku i od końca.

2. Analiza i specyfikacja

Do zrealizowania programu postanowiłem wykorzystać język C++, natomiast do wizualizacji wyników w formie wykresów użyłem zewnętrznej biblioteki graficznej SDL2.
W funkcjach przeze mnie napisnych potęgę i silnię obliczam w jednej pętli, aby skrócić czas wykonywania programu. Utworzyłem również tablicę silni n! dla n z przedziału od 0 do 99, którą uzupełniam tylko raz.

3. Kod źródłowy

Z powodu wykorzystania przeze mnie biblioteki graficznej wytworzonego przeze mnie kodu było dość sporo i wiele razy go zmieniałem, aby uzyskać odpowiednie wykresy. Zdecydowałem się więc zamieścić w załączniku tylko podglądowy program zawierający funkcje do obliczania funkcji $e^{\sin(x)}$ oraz ich proste użycie.

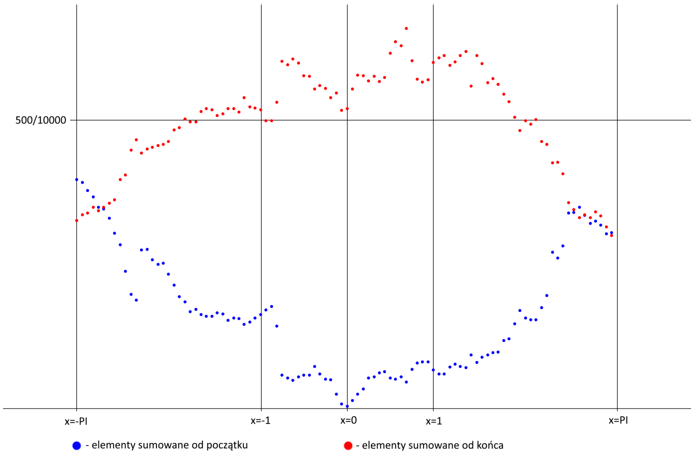
4. Hipotezy

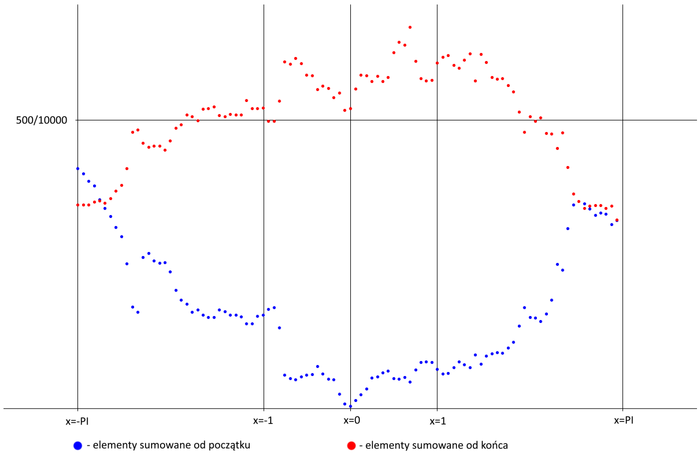
Testy do hipotez były przeprowadzane z następującymi ustawieniami:
- argumentów (x) było 1 000 000 i znajdywały się w równych odległościach w przedziale od –PI do +PI

- liczba elementów szeregu (n) dla H1 i H3 była stała i wynosiła 100, a dla H2: 8, 9 i 10.

Jeden punkt na wykresie oznacza ile razy dany sposób był lepszy od drugiego w przedziale 10 000 kolejnych po sobie x'ów (łącznie 100 punktów). Lepszy - czyli kiedy różnica z biblioteczną funkcją była mniejsza od przeciwnego sposobu.

- H1: sumowanie od końca daje dokładniejsze wyniki niż sumowanie od początku. Na potwierdzenie tej hipotezy zamieszczam następujące wykresy:





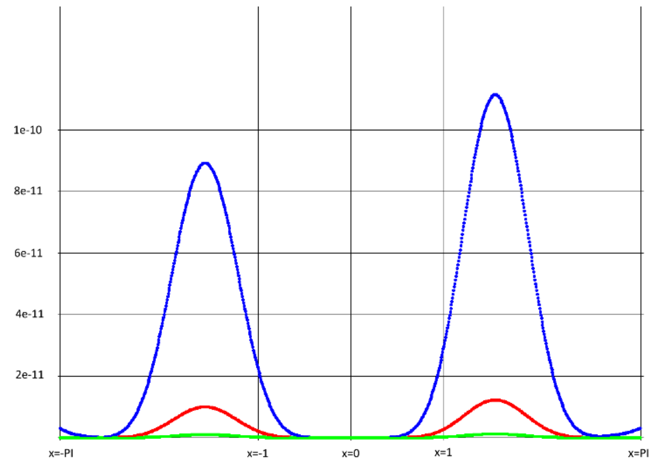
Elementy obliczane ze wzoru

Elementy obliczane na podstawie poprzedniego

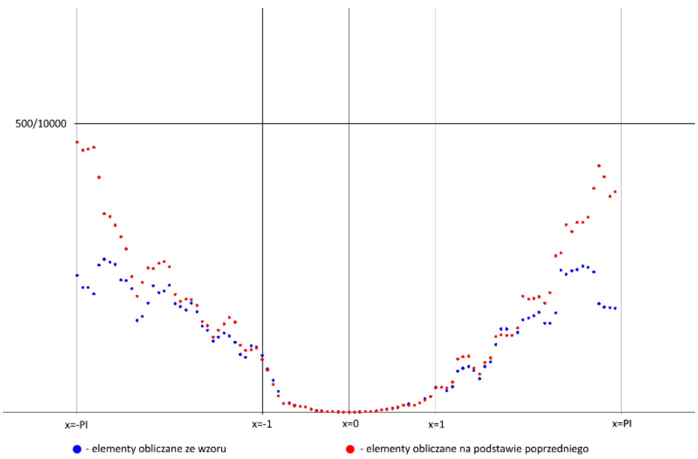
W obu przypadkach sumowanie od końca daje lepsze rezultaty.

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

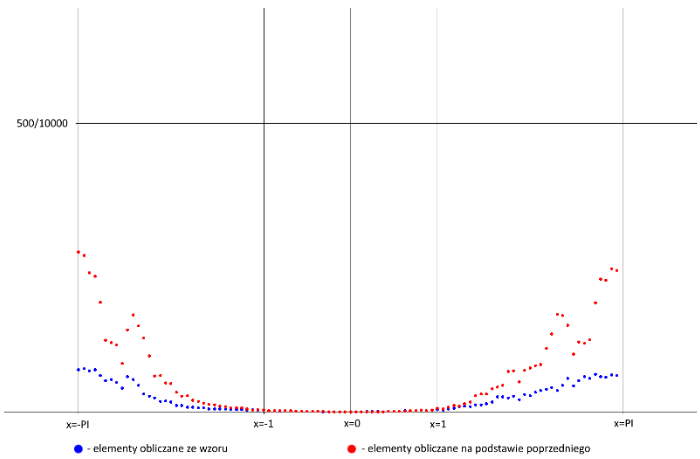
  Poniższy wykres przedstawia zależność błędu od argumentu x. Niebieski kolor dotyczny 8 sumowanych elementów, czerwony 9, a zielony 10. Jak widać dla każdej ilości sumowanych elementów wokół zera jest najdokładniejszy wynik, co potwierdza tą hipotezę. Ten test wykonałem sposobem, gdzie składniki szeregu obliczane są na podstawie poprzedniego i sumowane w kolejności od końca. Znajdują się tutaj również uśrednione wyniki testów.



- H3: sumowanie elementów obliczanych na podstawie poprzedniego daje dokładniejsze wyniki niż obliczanych bezpośrednio ze wzoru. Na potwierdzenie tej hipotezy zamieszczam następujące wykresy:



Elementy sumowane od początku



Elementy sumowane od końca

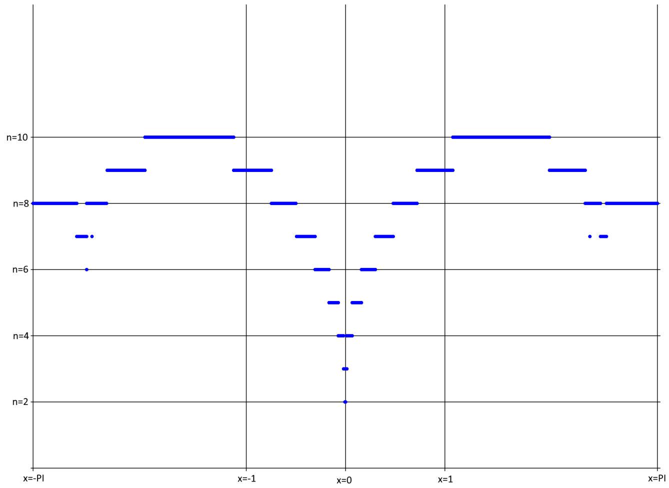
W obu przypadkach widać, że obliczanie kolejnych elementów na podstawie poprzedniego składnika daje dokładniejsze wyniki.

5. Pytania

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

  Na wykresie z hipotezy H2 umieściłem różne ilości sumowanych składników aby można było zauważyć, że im więcej tych elementów, tym błąd jest mniejszy, a więc wynik dokładniejszy.

- Q2: Ile składników należy sumować, aby otrzymać dokładność 10^{-6} w zależności od argumentu Odpowiedź do tego pytania przedstawiona jest na wykresie:



Według testu z uśrednionymi wynikami, dla x z przedziału od -PI do PI największą wymaganą liczbą elementów jest 10, natomiast w okolicach x=0 wystarczą nawet 2 elementy szeregu.

(dla x dostatecznie zbliżonego do 0, najprawdopodobniej wystarczyłby jeden element, ale na wykresie widać tylko uśrednione wyniki)