Szymon Urbański, Łukasz Stępień 2.03.2023r.

„Laboratorium 01”

Temat laboratorium: „Analiza błędów”

I. Zadanie 1

1. Temat zadania: obliczyć przybliżoną wartość pochodnej funkcji *tan(x)* w *x=1* używając wzorów:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (1)$$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \quad (2)$$

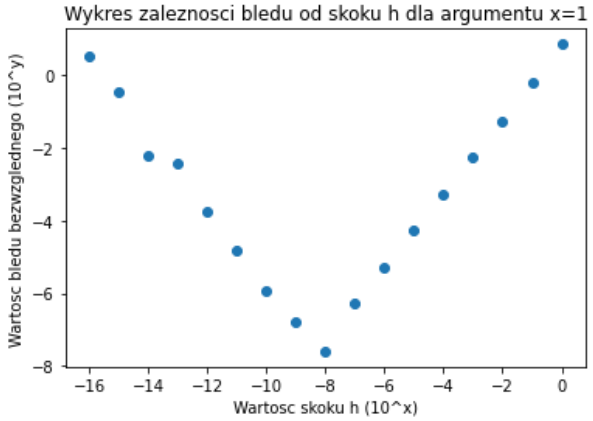
2. Implementacja funkcji:

- f_prim_est1(x,h) -> wzór (1),
- f_prim_est2(x,h) -> wzór (2),
- f_prim_ex(x) -> funkcja pochodna wyznaczona analitycznie (tan'(x)=1+tg^2(x)),
- zad1(f,x) -> główna funkcja rozwiązująca zadanie, przyjmuje funkcje reprezentującą wzór 1 lub 2 oraz argument dla tej funkcji.

3. Opis programu:

W pętli *for* wyliczane są przybliżone wartości funkcji *tan(x)* w punkcie x dla różnych wartości h= 10^{-k} dla k = 0,…,16. Wyniki są zapisywane w tablicy T w postaci krotek: (wartość h, błąd bezwzględny pomiędzy wartością dokładną a przybliżoną) ze skalą logarytmiczną. Następnie przygotowywany jest wykres reprezentujący wyniki obliczeń, wyliczana i wypisywana minimalna wartość wykresu oraz wartość epsilonu maszynowego.

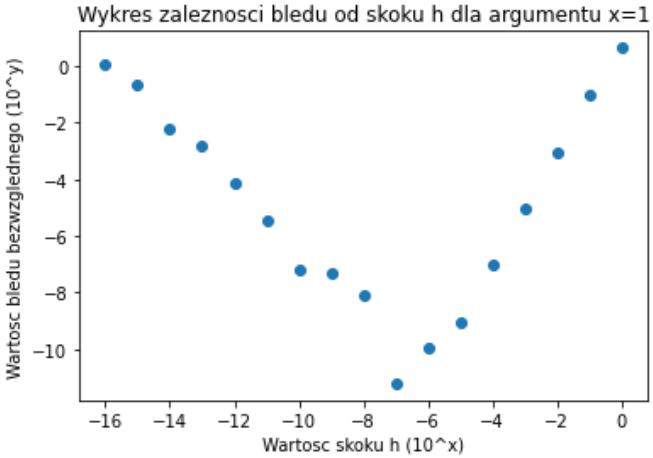
4.1. Wyniki dla wzoru (1):



Wykres 1.1 Wartości dla wzoru (1)

h_{min}	**1e-08**
$\sqrt{\epsilon_{mach}}$	**1.4901161193847656e-08**

4.2. Wyniki dla wzoru (2):



Wykres 1.2 Wyniki dla wzoru (2)

h_{min}	**1e-07**
$\sqrt{\epsilon_{mach}}$	**1.4901161193847656e-08**

5. Wnioski:

Algorytm wyliczania pochodnej ze wzorów (1) i (2) jest poprawny, co widać na wykresach błędu bezwzględnego. Posiadają one w obu przypadkach minimum, które jest zbliżone do pierwiastka z epsilonu maszynowego. Warto jednak zwrócić uwagę, że w zależności od przyjętej funkcji obliczania pochodnej otrzymano różne wartości h_{min}. Jest to związanie z tym, że pochodna centralna ma mniejszy błąd niż pochodna prawostronna. Można zatem stwierdzić, że w zależności od przyjętego wzoru zmienia się wartość *h* dla którego błąd obliczeniowy związany z reprezentacją liczb w komputerze jest najmniejszy, jednak dalej jest on zbliżony do pierwiastka ze słowa maszynowego.

II. Zadanie 2

1. Temat zadania: Wygenerować pierwsze wyrazy ciągu zdefiniowanego następująco:

$$x_{k+1} = 2.25x_k - 0.5x_{k-1}$$

o początkowych wyrazach $x_1 = \frac{1}{3}$, $x_2 = \frac{1}{12}$. Wykonać zadanie dla pojedynczej i podwójnej precyzji reprezentacji liczb w komputerze. Narysować kolejne wyrazy ciągu oraz porównać je z rzeczywistymi wartościami.

2. Implementacja funkcji:

- - generate1(n) -> funkcja generująca *n* wyrazów ciągu z pojedynczą precyzją (liczby zapisane na 32 bitach)
- - generate2(n) -> funkcja generująca *n* wyrazów ciągu z pojedynczą precyzją (liczby zapisane na 64 bitach)

- - plot(T) -> funkcja ustawiająca wykres danych z tablicy T
- - zad2(n1, n2) -> główna funkcja programu

3. Opis programu.

Główna funkcja programu na początku generuje wartości ciągu dla pojedynczej (60 wyrazów) i podwójnej precyzji (225 wyrazów) oraz dokładne wartości ciągu na podstawie podanego wzoru:

$$x_k = \frac{4^{1-k}}{3}.$$

Następnie obliczane są bezwzględne błędu dla obu przypadków i rysowane są wykresy:

- wykres wartości ciągu dla pojedynczej precyzji

- wykres wartości ciągu dla podwójnej precyzji

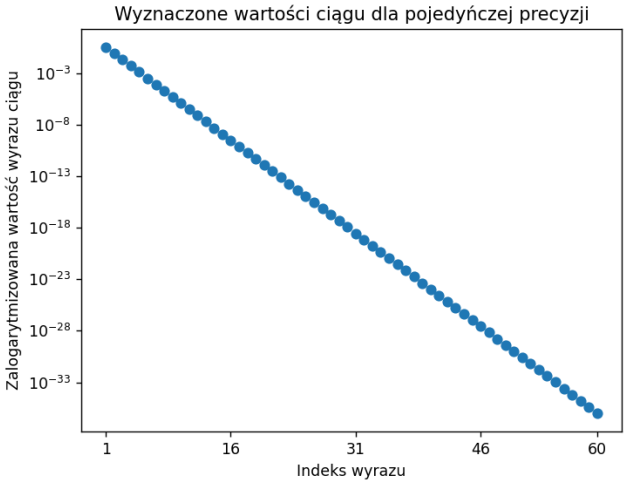
- wykres rzeczywistych wartości ciągów

- wykres błędu bezwzględnego dla pojedynczej precyzji

- wykres błędu bezwzględnego dla podwójnej precyzji

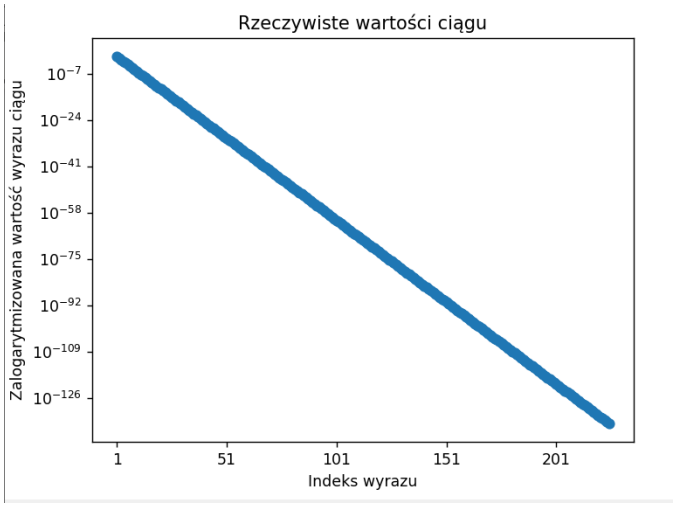
4. Wyniki

Wyniki przedstawiono na poniższych wykresach.

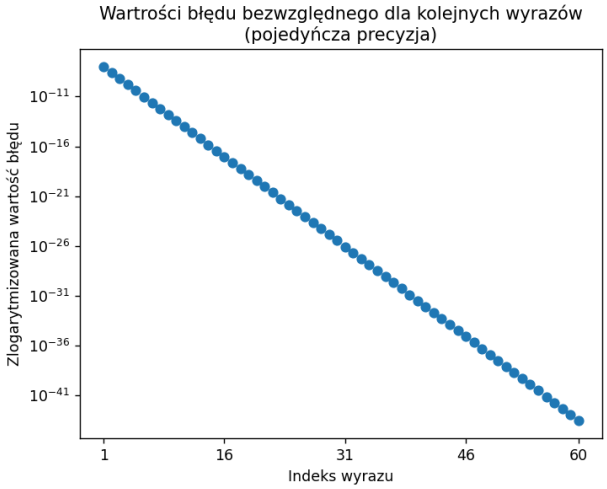


Wykres 2.3 Wartości ciągu dal pojedynczej precyzji

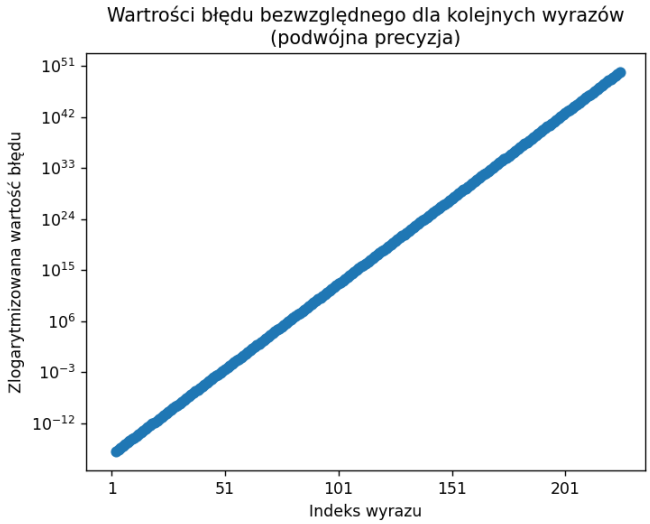
Wykres 2.2 Wartości ciągu wyznaczone dla podwójnej precyzji.



Wykres 2.3 Rzeczywiste wartości ciągu.



Wykres 2.4 Wartości błędu dla pojedynczej precyzji.



Wykres 2.5 Wartości błędu dla podwójnej precyzji.

5. Wnioski.

Jak widać na powyższych wykresach dla początkowych wyrazów ciągu lepsza okazała się pojedyncza precyzja. Może to wynikać z faktu, że kolejne wyrazy są położone bardzo blisko siebie i błędy zaokrągleń bardziej nawarstwiają się w podwójnej precyzji. Dodatkowo warto zauważyć, że w podwójnej precyzji w pewnym momencie ciąg zaczyna być rosnący, mimo że teoretycznie powinien być malejący. Dzieje się tak, ponieważ błędy przybliżeń numerycznych nakładają się na siebie i w pewnym momencie kolejne wyrazy zaczynają być większe od poprzednich. Można również zauważyć, że dla pojedynczej precyzji wartość błędu maleje, natomiast dla podwójnej precyzji wartość błędu wzrasta wraz ze wzrostem indeksu. Jest to następstwem tego, że dla początkowych wyrazów ciągu pojedyncza precyzja okazuję się być bardziej skuteczna generuje ciąg malejący (zgodnie z teoretycznym założeniem). Warto również zwrócić uwagę, że dla podwójnej precyzji wartość błędu sięga nawet wartości 10^{50}, przy czym rzeczywisty wynik jest rzędu 10^{-126}. Oznacza to, że użyte przybliżenia spowodowały drastyczną zmianę wyniku (w szczególności to, że w pewnym momencie ciąg zaczął rosnąć). Nie można jednak jednoznacznie stwierdzić, że pojedyncza precyzja jest lepsza dla całego ciągu, gdyż zbadano tylko 60 początkowych wyrazów. Możliwe, że gdyby badano kolejne wyrazy ciągu nastąpiłaby podobna sytuacja jak w podwójnej precyzji i ciąg zacząłby rosnąć. Obliczając kolejne wyrazy ciągu rekurencyjnego należy pamiętać, że błąd wyznaczenia kolejnych wyrazów jest nie tylko błędem wynikającym z ograniczeń reprezentacji liczb w komputerze, ale także następstwem wcześniej dokonanych przybliżeń. Dlatego też przy rekurencyjnym wyznaczaniu liczb zmiennoprzecinkowych trzeba pamiętać, że kolejne wyrazy mogą mieć coraz większe błędy i że te błędy mogą powiększać się w coraz szybszym tempie.

Bibliografia:

- Qingkai Kong, Timmy Siauw, Alexandre Bayen - *Python Programming and Numerical Methods*