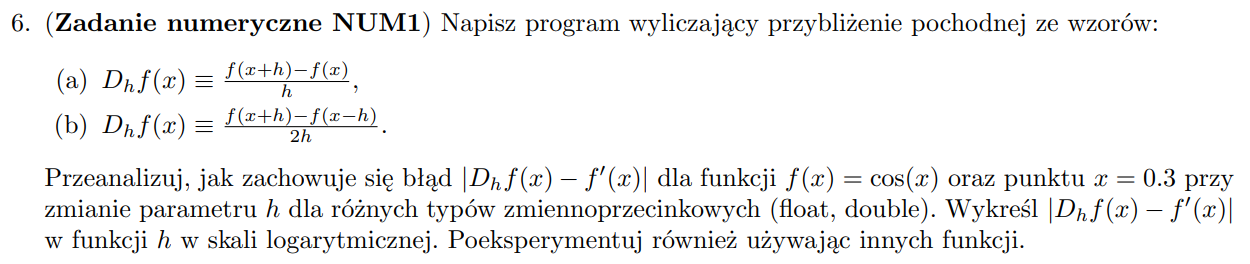
# Zadanie numeryczne 1



# Wprowadzenie

Program liczy przybliżenie pochodnej funkcji w punkcie przez podstawienie odpowiednio małej wartości do wzoru który normalnie liczy granicę dla .

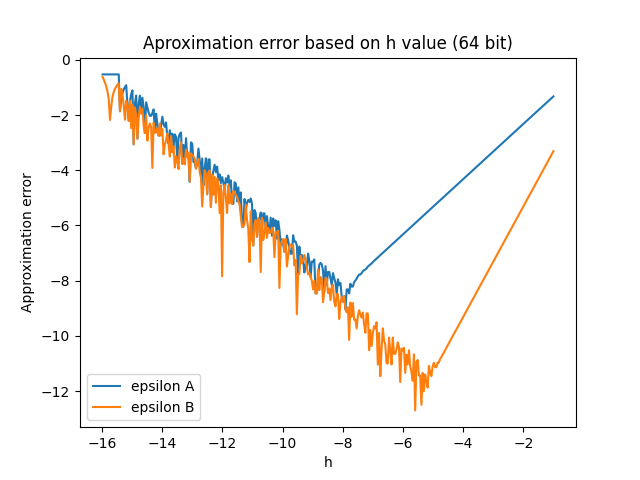
Następnie program oblicza błąd wyliczonego przybliżenia, przez porównanie go z rzeczywistą wartością pochodnej w punkcie oraz rysuje wykres pokazujący jak zmienia się otrzymany błąd obliczeniowy w zależności od wybranego .

# Wynik

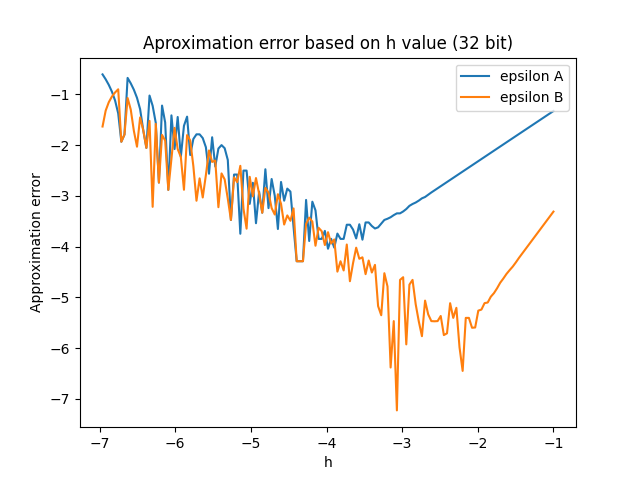
Do rozwiązania zadania użyłem dwóch programów, które w działaniu są prawie identyczne, z tym, że program32.py operuje na 32 bitowych liczbach i zakresie , a program64.py na 64 bitowych liczbach i zakresie .

Programy te obliczają epsilon dla obu wzorów przez obliczenie wartości bezwzględnej z różnicy przybliżenia danego wzorem oraz rzeczywistej wartości pochodnej funkcji .

Do obliczeń na typach zmiennoprzecinkowych została wykorzystana biblioteka numpy, a do wygenerowania wykresów matplotlib.

**Wykres otrzymany dla działań na 64-bitowych liczbach:** 

**Wykres otrzymany dla działań na 32-bitowych liczbach:**

­

# Dyskusja wyników

Otrzymany wykres błędu dla równania a przypomina literę V, ponieważ w miarę jak zmniejszamy h, zbliżamy się do 0 co precyzyjniej odpowiada oczekiwanej granicy ze wzoru – stąd widać jak na prawej stronie wykresu płynnie zmniejsza się błąd. Jednak we wzorze występuje również dzielenie przez h, co sprawia że gdy h jest bardzo małe, błędy spowodowane przez określoną precyzję działań zmiennoprzecinkowych na komputerze są bardzo mocno powielone. Z tego powodu po pewnym optymalnym , wzięcie mniejszego h powoduje wzrost błędu, co powoduje nieregularną lewą stronę wykresu, na której widać powielone błędy obliczeniowe.

Wykres błędu dla równania b osiąga lepszą maksymalną precyzję, co jest spowodowane użyciem innego wzoru na przybliżenie, w którym dzielimy przez liczbę , co redukuje błędy spowodowane dzieleniem przez małą liczbę.

W przypadku działań na 32 bitowych typach zmiennoprzecinkowych, wykresy mają podobne kształty, jednak znacznie mniejszą maksymalną precyzję.

Dla liczb 64 bitowych otrzymane wartości to:

1. ,

A dla liczb 32 bitowych:

1. ,