

Definicje funkcji:

- f32, f64: Funkcje sinus oparte na danych typu float32 i float64, odpowiednio.
- g32, g64: Funkcje wykładnicze dla bazy 3 również na danych typu float32 i float64.
- fPrim32, fPrim64: Analityczne pochodne funkcji sinus.
- gPrim32, gPrim64: Analityczne pochodne funkcji wykładniczej.

Obliczanie pochodnych:

- derivative_a32, derivative_a64: Numeryczne obliczanie pochodnej metodą różnicy w przód dla danych float32 i float64.
- derivative_b32, derivative_b64: Numeryczne obliczanie pochodnej metodą różnicy centralnej dla danych float32 i float64.

Kod bada błędy popełniane podczas obliczania pochodnych numerycznych, porównując wyniki numeryczne z dokładnymi pochodnymi analitycznymi. Błędy są obliczane jako różnica między aproksymowaną pochodną a pochodną dokładną.

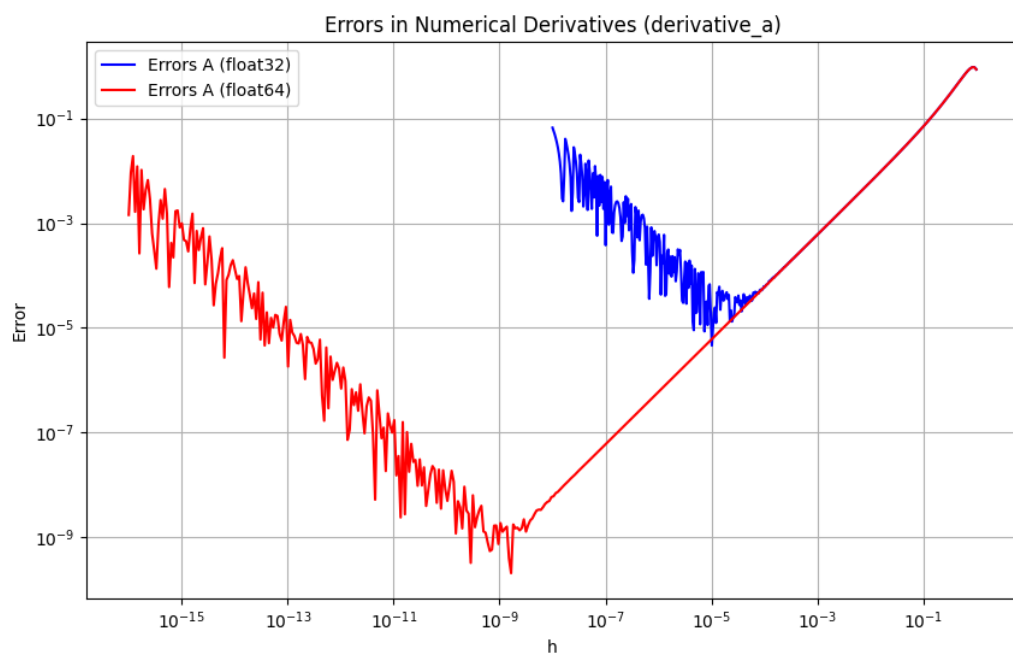
W analizie wykorzystano:

- **Zmienną krokową h** , która była generowana przy użyciu funkcji np. `logspace`. Dla danych float32 zakres kroków wynosił od 10^{-8} do 1, a dla float64 od 10^{-16} do 1.
- Dwie różne metody obliczania pochodnych:
 - **Różnica w przód (metoda A):** $D_h f(x) \equiv (f(x + h) - f(x))/h$
 - **Różnica centralna (metoda B):** $D_h f(x) \equiv (f(x + h) - f(x - h))/(2h)$

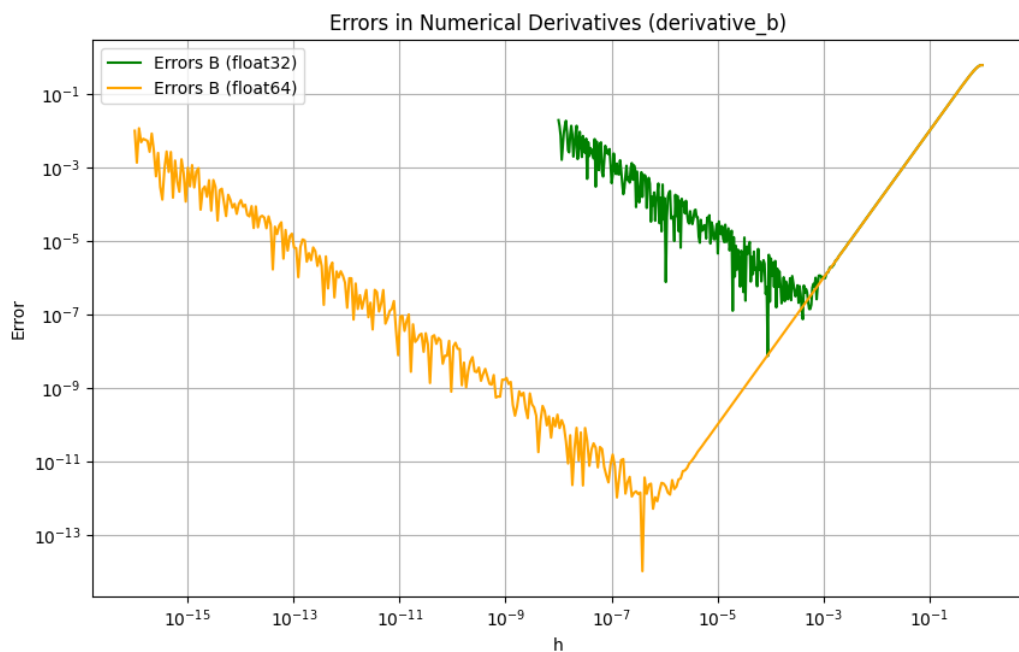
Błędy dla każdej metody i każdej funkcji zostały zwizualizowane na wykresach w skali logarytmicznej dla zmiennej krokowej h i błędu pochodnej.

W kodzie wygenerowano cztery wykresy:

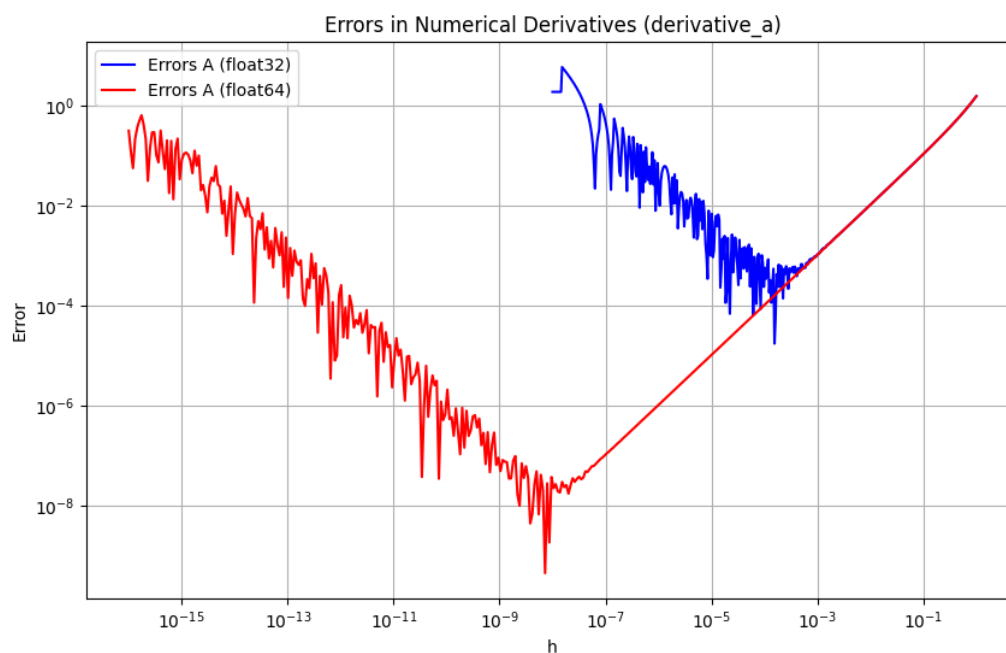
1. **Błędy dla funkcji sinus (derivative_a)**, z zastosowaniem metody różnicy w przód dla typów danych float32 i float64.



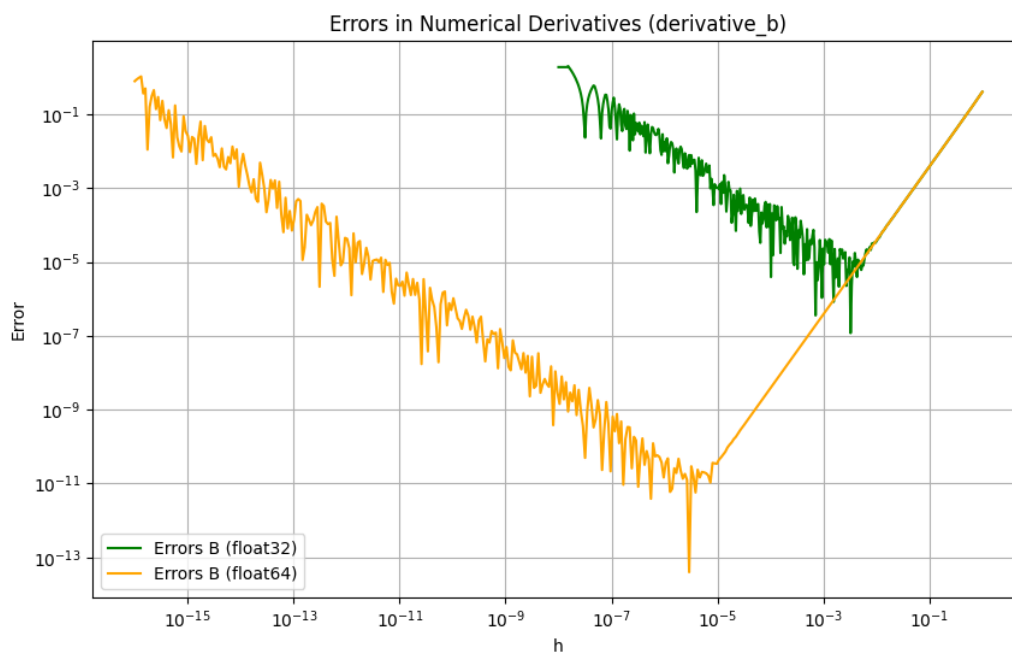
2. **Błędy dla funkcji sinus (derivative_b), z zastosowaniem metody różnicy centralnej.**



3. **Błędy dla funkcji wykładniczej (derivative_a).**



4. Błędy dla funkcji wykładniczej (derivative_b).



Wszystkie wykresy przedstawiono na skalach log, co pozwala na lepsze zrozumienie, jak krok różniczkowania h wpływa na dokładność aproksymacji.

Obie funkcje reagują podobnie na zmiany kroku h , ale ze względu na swój różny charakter matematyczny, funkcja sinus generuje większe błędy, szczególnie przy małych krokach i przy użyciu mniej precyzyjnych danych float32. Funkcja wykładnicza

wykazuje lepszą stabilność numeryczną i mniejsze błędy, zwłaszcza przy zastosowaniu metody różnicy centralnej i wyższej precyzji float64.

Z analizy wyników można wyciągnąć kilka wniosków:

1. **Precyzja a dokładność:** Typ danych float64 jest bardziej dokładny niż float32, co jest widoczne szczególnie przy bardzo małych krokach h . Przy mniejszych krokach h , liczby zmiennoprzecinkowe float32 są mniej precyzyjne, co prowadzi do większych błędów numerycznych.
2. **Wpływ kroku h :** Błędy rosną dla bardzo małych wartości h z powodu problemów numerycznych (np. zaokrągleń) oraz dla zbyt dużych kroków, gdzie aproksymacja staje się niedokładna.
3. **Porównanie metod:** Metoda różnicy centralnej (derivative_b) jest bardziej dokładna niż metoda różnicy w przód (derivative_a), co wynika z jej wyższego rzędu dokładności. W metodzie różnicy w przód, dominują błędy wynikające z zaokrągleń i małego kroku, podczas gdy w różnicy centralnej te błędy są mniejsze.
4. **Typ danych:** Chociaż float64 oferuje większą precyzję, to nie zawsze jest konieczna – w wielu przypadkach float32 daje wystarczająco dokładne wyniki przy odpowiednio dobranym kroku h .

Podsumowując, dobór odpowiedniego kroku h oraz stosowanie bardziej zaawansowanych metod, takich jak różnica centralna, znacząco poprawia dokładność numerycznego różniczkowania, nawet przy ograniczonej precyzji, jak w przypadku float32. Metoda różnicy centralnej, dzięki swojej wyższej dokładności i symetrycznemu podejściu, lepiej radzi sobie z błędami zaokrągleń niż metoda różnicy w przód. Odpowiednia kombinacja metody i precyzyjnie wybranego kroku pozwala na uzyskanie stabilniejszych i bardziej dokładnych wyników.