Zbiór Mandelbrota na GPU

# Wstep historyczny

Benoît B. Mandelbrot był fancuskim i amerykańskim [matematyk](https://pl.wikipedia.org/wiki/Matematyka)iem [polsko-żydowskiego](https://pl.wikipedia.org/wiki/%C5%BBydzi_w_Polsce) pochodzenia. Urodził się [20 listopada](https://pl.wikipedia.org/wiki/20_listopada) [1924](https://pl.wikipedia.org/wiki/1924) w [Warszawie](https://pl.wikipedia.org/wiki/Warszawa). Zmarł [14 października](https://pl.wikipedia.org/wiki/14_pa%C5%BAdziernika) [2010](https://pl.wikipedia.org/wiki/2010) w [Cambridge (Massachusetts)](https://pl.wikipedia.org/wiki/Cambridge_(Massachusetts)). Zajmował się szerokim zakresem problemów matematycznych, znany jest przede wszystkim jako ojciec geometrii fraktalnej, opisał [zbiór Mandelbrota](https://pl.wikipedia.org/wiki/Zbi%C3%B3r_Mandelbrota) oraz wymyślił samo słowo „[fraktal](https://pl.wikipedia.org/wiki/Fraktal)”.

# Czym jest Fraktal?

Fraktal to pewien obiekt, który w sensie geometrycznym jest samopodobny, czyli jakaś jego część jest podobna do całości.

Ważnym fraktalem jakim będziemy zajmowali się w tej pracy jest brzeg Zbioru Mandelbrota. Jest to specyficzny fraktal który istnieje tylko w świecie matematycznym. Sam Zbiór Mandelbrota jest podzbiorem [płaszczyzny zespolonej](https://pl.wikipedia.org/wiki/P%C5%82aszczyzna_zespolona) i nie jest fraktalem, ponieważ nie jest samopodobny, co zostało dowiedzione przez chińską matematyczkę [Tan Lei](https://pl.wikipedia.org/w/index.php?title=Tan_Lei&action=edit&redlink=1).

# Wzór na Zbiór Mandelbrota

Nasz [zbiór](https://pl.wikipedia.org/wiki/Zbi%C3%B3r) możemy opisać jako te punkty p(zespolone) p ∈ C , {\displaystyle p\in \mathbb {C} ,} dla których [ciąg](https://pl.wikipedia.org/wiki/Ci%C4%85g_(matematyka)) (Zi)infi=0 ( z i ) i = 0 ∞ {\displaystyle (z\_{i})\_{i=0}^{\infty }} zdefiniowany [równaniem rekurencyjnym](https://pl.wikipedia.org/wiki/R%C3%B3wnanie_rekurencyjne)



nie dąży do [nieskończoności](https://pl.wikipedia.org/wiki/Niesko%C5%84czono%C5%9B%C4%87)



Można wykazać, że jest to równoważne z



Podsumowując jednym zdaniem:



# Część praktyczna

W mojej pracy postanowiłem przedstawić porównanie trzech algorytmów tworzących zbiór Mandelbrota:

1. Na CPU przy pomocy jednego wątku
2. Na CPU przy pomocy wielu wątków
3. Na GPU

Porównanie działania tych algorytmów omówię w kolejnej części. Tutaj chciałbym się skupić na głównych różnicach w kodzie programów oraz statystycznym porównaniu ich czasów działania.

Zwrócimy także uwagę na konkretne fragmenty kodu odpowiadające za te różnice i omówimy co za to odpowiada.

Podstawowym aspektem przemawiającym za przewagą GPU nad CPU w tym typie problemu jest fakt wyspecjalizowania do intensywnych obliczeniowo, wysoce równoległych obliczeń bycia zaprojektowanym w taki sposób, by więcej tranzystorów było przeznaczonych do przetwarzania danych, a nie do buforowania danych i kontroli przepływu.

# Porównanie czasowe

W tym miejscu tak jak zapowiadałem porównam faktyczne różnice czasowe dla trzech zastosowanych algorytmów.

Jak widać na wykresie zdecydowanie możemy uznać, że użycie GPU jest o wiele lepszy sposobem na przybliżanie zbioru Mandelbrota.

Użycie wielu wątków zdecydowanie może przyspieszyć czas działania programu, lecz nie daje nam to nadal optymalnych wyników.

Pomimo iż algorytm jednowątkowy od początku zdawał się zbyt powolny po testach możemy stwierdzić, że jest możliwe używanie go przy obliczaniu Zbioru Mandelbrota, choć nie daje nam on wyników w szybkim czasie to nadal nie musimy czekać wieczności i jesteśmy pewni, że uzyskane wyniki nadal będą poprawne.

# Podsumowanie

Biorąc pod uwagę trzy algorytmy, których użyłem na potrzeby tej pracy możemy jednozancznie stwierdzić, że prędkość obliczeniaowa GPU jest zdecydowanym atutem przeważającym nad nawet wielowątkowym CPU. Pomimo ogromnej ilości wygenerowanych pikseli GPU dalej utrzymuje swój czas pracy na niskim poziomie i potwierdza nasze wcześniejsze spekulacje.

Użycie wielu wątków na CPU skraca czas obliczeń w znacznym stopniu (o ponad połowe), lecz nadal nie dorównuje to prędkości GPU.

Procesor jednowątkowy jest w stanie wygenerować obraz w zadowalającym czasie, ale nie jest w tym przypadku zalecany z powodu swojej szybko rosnącej zależności czasowej od ilości generowanych pikseli.

Wybranie GPU w tym przypadku jest słuszną i uzasadnioną decyzją potwierdzoną dokładną analizą i testami.