

# Procesory graficzne w obliczeniach równoległych (CUDA)

## Lista nr 2: atraktory i fraktale

### Zadanie 1:

Wielomian zespolony  $z^4 - 1 = 0$  ma cztery rozwiązania:  $1, i, -1, -i$ . Szukając zer metoda Newtona iterujemy kolejne przybliżenia:  $z_{n+1} = (3z_n + z_n^{-3})/4$ . Każdy z punktów płaszczyzny z którego zaczniemy iteracje wyląduje w jednym z rozwiązań (epsilonowym otoczeniu) lub nie. W zależności od tego należy przypisać punktom płaszczyzny jeden z 5 kolorów.

Zaimplementować funkcje na CPU i GPU, które wygenerują tablice kolorów dla obszaru płaszczyzny  $[-s, s] \times [-s, s]$  (np.  $s = 2$ ) i rozdzielczości M kolumn i N wierszy:

```
char C[M*N]; // tablica na wartości kolorów: 1-5
void CPUAtraktor(char *C, int M, int N, float s);
void GPUAtraktor(char *C, int M, int N, float s);
```

Program testujący może wypisywać na terminalu tablice kolorów jako ASCII-ART w postaci 5 różnych znaków "oxXW" lub zapisywać obraz do pliku graficznego (można użyć zewnętrznej biblioteki). Na wyjściu powinny być wypisane informacje o czasach wykonania na CPU i GPU i sprawdzenie czy rezultaty są takie same. Jako pierwsze argumenty wywołania programu podajemy rozdzielczość: M N.

Zaimplementuj jak najwięcej funkcji wspólnych dla CPU i GPU np. obliczanie następnego wyrazu ciągu lub operacje na liczbach zespolonych np. dodawanie:

```
__host__ __device__ void cadd(float *a, float *b, float *c){...}
```

### Uwagi:

1. Dodawanie liczb zespolonych i mnożenie każdy zna. Odwrotność dla liczb o normie 1 to liczba sprzężona:  $(a + bi)(a - bi) = a^2 + b^2 = 1$ . A jeśli norma liczby inna to wystarczy ją unormować aby mieć wzór na  $1/z$ .
2. W prostszej wersji można przyjąć, że  $M, N \leq 512$ .

Andrzej Łukaszewski