

# Transformacja Fouriera i algorytm FFT

Michał Saturczak

Informatyka Stosowana, Akademia Górniczo-Hutnicza

Metody Numeryczne

2 czerwca 2024

# 1 Wstęp teoretyczny

Transformacja Fouriera (FT) jest fundamentalnym narzędziem w analizie sygnałów i przetwarzaniu danych. Umożliwia ona przejście z dziedziny czasu (lub przestrzeni) do dziedziny częstotliwości, dostarczając informacji o składowych sinusoidalnych sygnału. W praktyce, dyskretna transformacja Fouriera (DFT) jest stosowana do sygnałów dyskretnych, takich jak te reprezentowane w komputerach.

DFT jest obliczeniowo kosztowna, szczególnie dla dużych zbiorów danych. Szybka Transformata Fouriera (FFT) jest rodziną algorytmów, które znacznie przyspieszają obliczenia DFT, redukując złożoność obliczeniową z  $(O(N^2))$  do  $(O(N \log N))$ , gdzie  $(N)$  jest liczbą próbek sygnału.

FFT znajduje szerokie zastosowanie w różnych dziedzinach, takich jak:

- Przetwarzanie sygnałów: analiza widmowa, filtracja, kompresja.
- Analiza obrazu: kompresja obrazu (JPEG), rozpoznawanie wzorców.
- Akustyka: analiza dźwięku, synteza mowy.
- Medycyna: obrazowanie medyczne (MRI, tomografia komputerowa).
- Fizyka: symulacje numeryczne, rozwiązywanie równań różniczkowych.

## 2 Problem

### 2.1 Rozwiązanie problemu

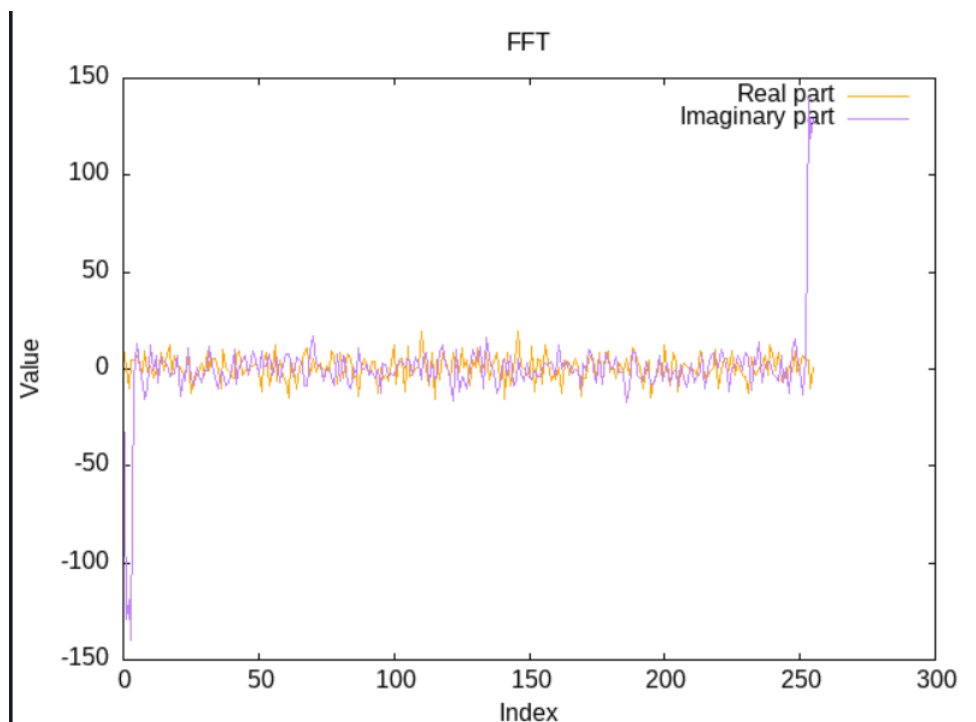
- Napisano program w języku C++ z do rozwiązania problemów związanych z Transformacją Fouriera i algorytmem FFT
- Wygenerowano i zapisano sygnał
- Obliczono transformację Fouriera i zapisano jego moduł do pliku CSV
- Obliczono odwrotną transformatę
- Zapis odsumionego sygnału
- Generowanie wykresów przy pomocy gnuplota

### 2.2 Funkcje użyte do rozwiązania problemu

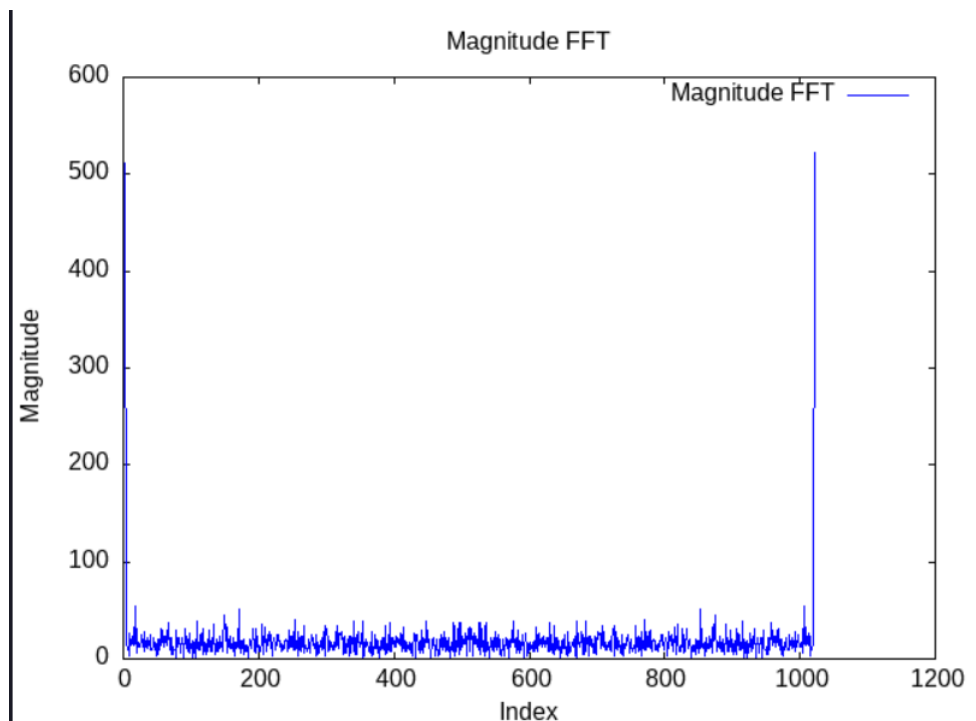
- `generate_signal(clean_signal, N, omega)` generuje czysty sygnał
- `save_to_csv(clean_filename, clean_signal, N)` zapisuje czysty sygnał do pliku CSV
- `generate_signal(signal, N, omega)` generuje sygnał, który będzie zaszumiony
- `add_noise(signal, N)` dodaje szum do sygnału
- `save_to_csv(noisy_filename, signal, N)` zapisuje zaszumiony sygnał do pliku CSV

### 3 Wyniki

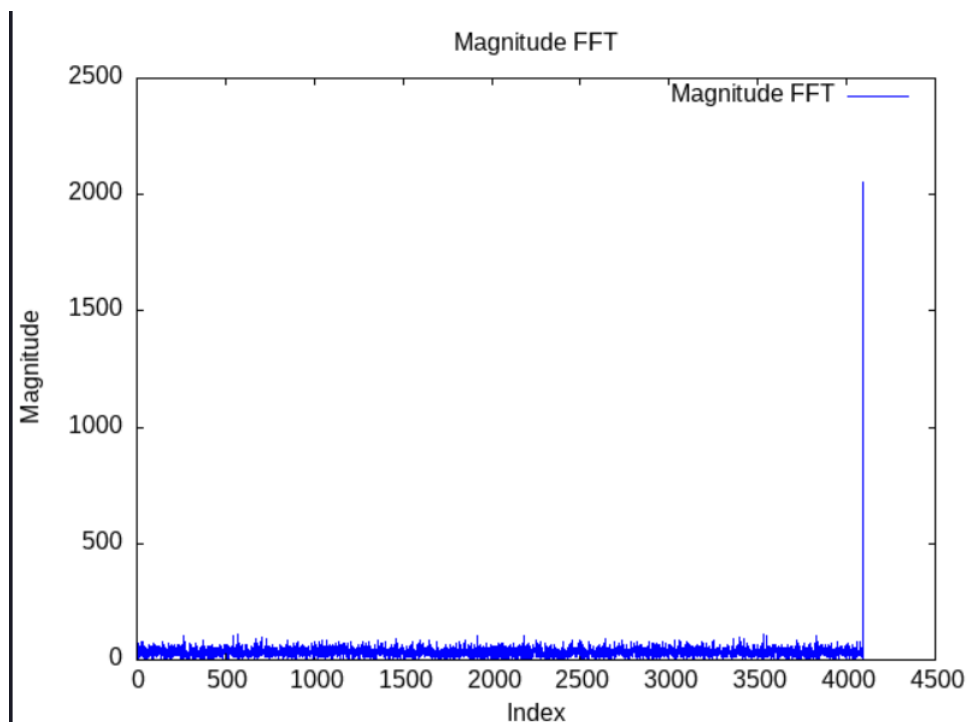
#### 3.1 Wykres części rzeczywistej i urojonej



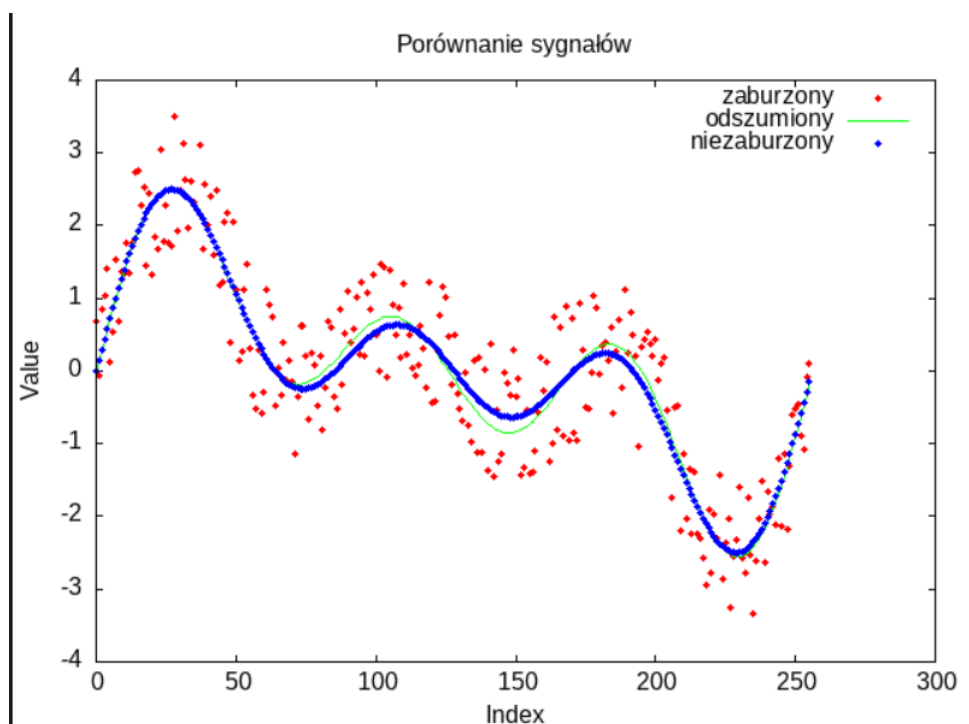
#### 3.2 Wykres dla widma amplitudowego sygnału $n = 10$



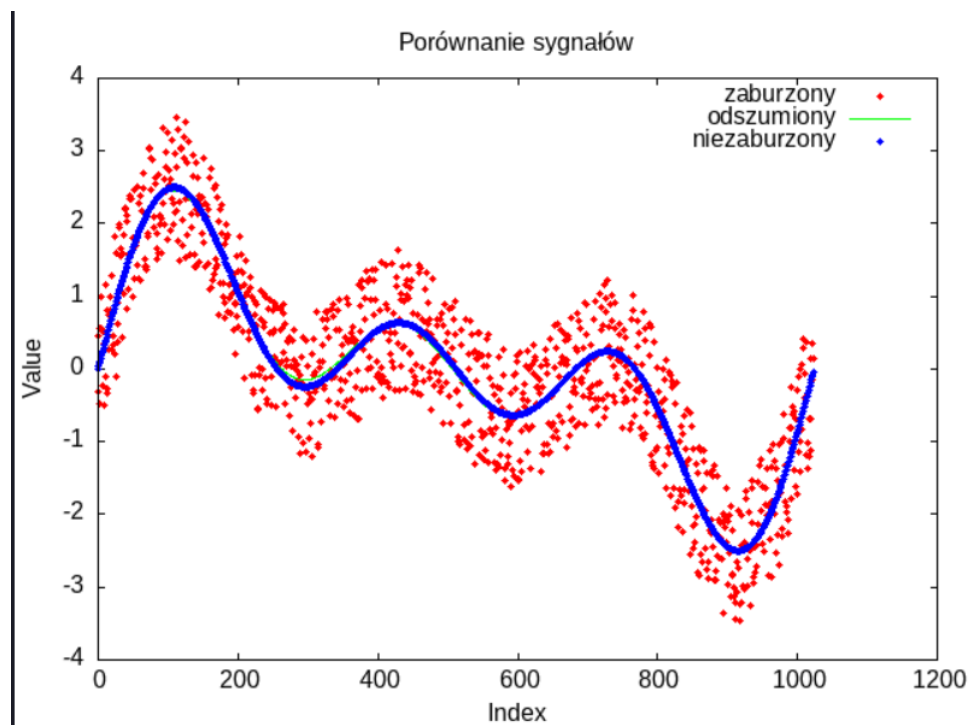
### 3.3 Wykres dla widma amplitudowego sygnału $n = 12$



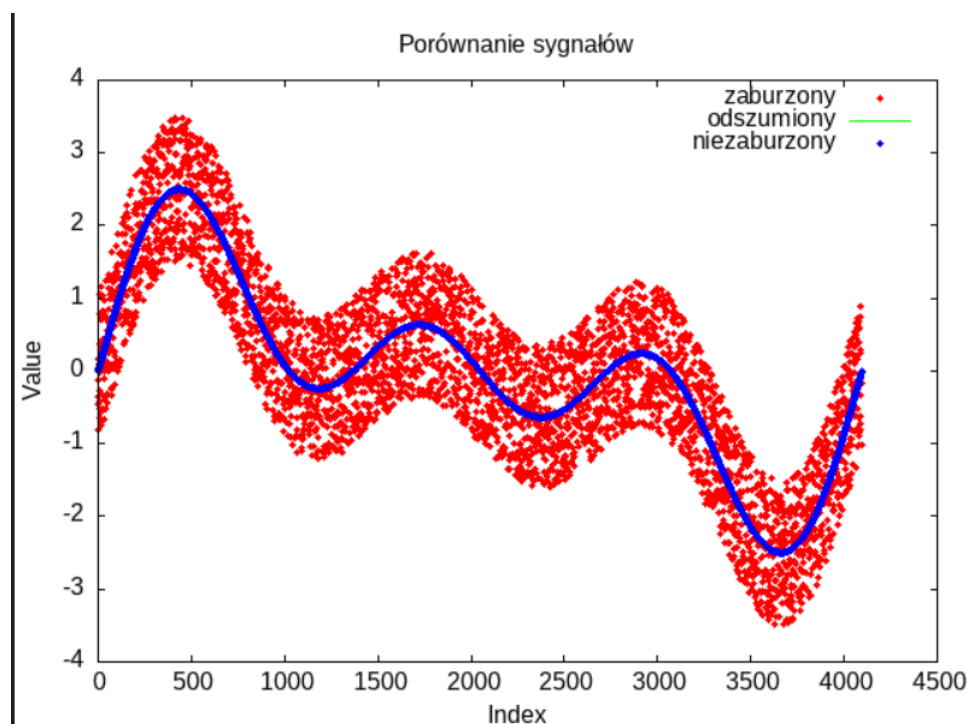
### 3.4 Wykres dla zaburzonego, niezaburzonego oraz odszumionego dla $n = 8$



3.5 Wykres dla zaburzonego, niezaburzonego oraz odszumionego dla  $n = 10$



3.6 Wykres dla zaburzonego, niezaburzonego oraz odszumionego dla  $n = 12$



## 4 Wnioski

Częstotliwość próbkowania ma istotny wpływ na wyniki transformacji Fouriera. Wyższa częstotliwość próbkowania prowadzi do dokładniejszej reprezentacji sygnału w dziedzinie częstotliwości, co jest kluczowe dla skutecznej analizy i przetwarzania.

Algorytm FFT umożliwia efektywne usuwanie szumu z sygnałów poprzez identyfikację i eliminację niepożądanych składowych częstotliwościowych. Dyskryminacja w dziedzinie częstotliwości, jak pokazano w przykładach, pozwala na skuteczną redukcję szumu, zachowując istotne cechy sygnału.