Vadym Semkovych
296669
20.05.2020

Sprawozdanie 11

Aproksymacja sygnału okresowego przy użyciu FFT

1. Wstęp teoretyczny

Szybka transformacja Fouriera (Fast Fourier Transform, FFT) – algorytm wyznaczania dyskretnej transformaty Fouriera oraz transformaty do niej odwrotnej.

Najpopularniejszą wersją algorytmu FFT jest FFT o podstawie 2. Jest on bardzo efektywny pod względem czasu realizacji, jednak wektor próbek wejściowych (spróbkowany sygnał) musi mieć długość $N = 2^k$ gdzie k to pewna liczba naturalna. Wynik otrzymuje się na drodze schematycznych przekształceń.

Najprostszy algorytm FFT to radix-2 (Cooley-Tukey). Algorytm polega na znalezieniu współczynników transformaty Fouriera (DFT) c_k, ale wykonując jak najmniej obliczeń.

Zakładamy, że całkowita liczba węzłów jest potęgą 2:

$$x_j = \frac{2\pi}{N} j,\ \ j = 0,1,2,\dots,N-1,\ \ N = 2^r, r \in \boldsymbol{N} \quad (1)$$

Zatem:

$$c_k = \sum_{j=0}^{N} f_j exp(-I\frac{2\pi}{N}jk) \quad (2)$$

Osobno grupujemy składniki:

Parzyste $j = 2m$:

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} exp(-I\frac{2\pi}{N}(2m)k) + \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} exp(-I\frac{2\pi}{N}(2m+1)k) \quad (3)$$

Niparzyste $j = 2m+1$:

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} exp(-I\frac{2\pi}{\frac{N}{2}}mk) + \exp(-I\frac{2\pi}{N}k) \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} exp(-I\frac{2\pi}{\frac{N}{2}}mk) \quad (4)$$

Korzystając z okresowości powyższych wyrazów nie musimy wyznaczać wszystkich współczynników – tylko połowę.

2. Problem

Naszym zadaniem było zastosowanie FFT do odszumienia sygnału periodycznego. Sygnał zaszumiony generowaliśmy zgodnie z poniższym wzorem:

$$y(i) = \sin(\omega \cdot i) + \sin(2\omega \cdot i) + \sin(3\omega \cdot i) + \Delta \qquad (5)$$

gdzie: $i = 0,1,2, \dots, N-1$ – numer próbki sygnału,
$\omega = 2\frac{2\pi}{N}$, $N = 2^k$ – ilość wygenerpwanych próbek,
$\Delta = 2 \cdot \frac{rand()}{RAND_{MAX}+1.0} - 1$

Najpierw wygenerowaliśmy zaszumiony sygnał, zgodnie ze wzorem(5), i zapisaliśmy go do wektora typu *double*. Długość wektora wynosiła 2N, $N = 2^k$, kolejno dla $k = 8, 10, 12$.
Następnie wyznaczyliśmy transformatę sygnału korzystając z funkcji biblioteki **GSL:**

gsl_fft_complex_radix2_forward (gsl_complex_packed_array y, size_t sride, size_t N),

gdzie: y – tablica z sygnałem(typ gsl_complex_packed_array odpowiada naszej jednowymiarowej tablicy),
stride – parametr równy 1(krok pomiędzy elementami w tablicy),
N – 2^k

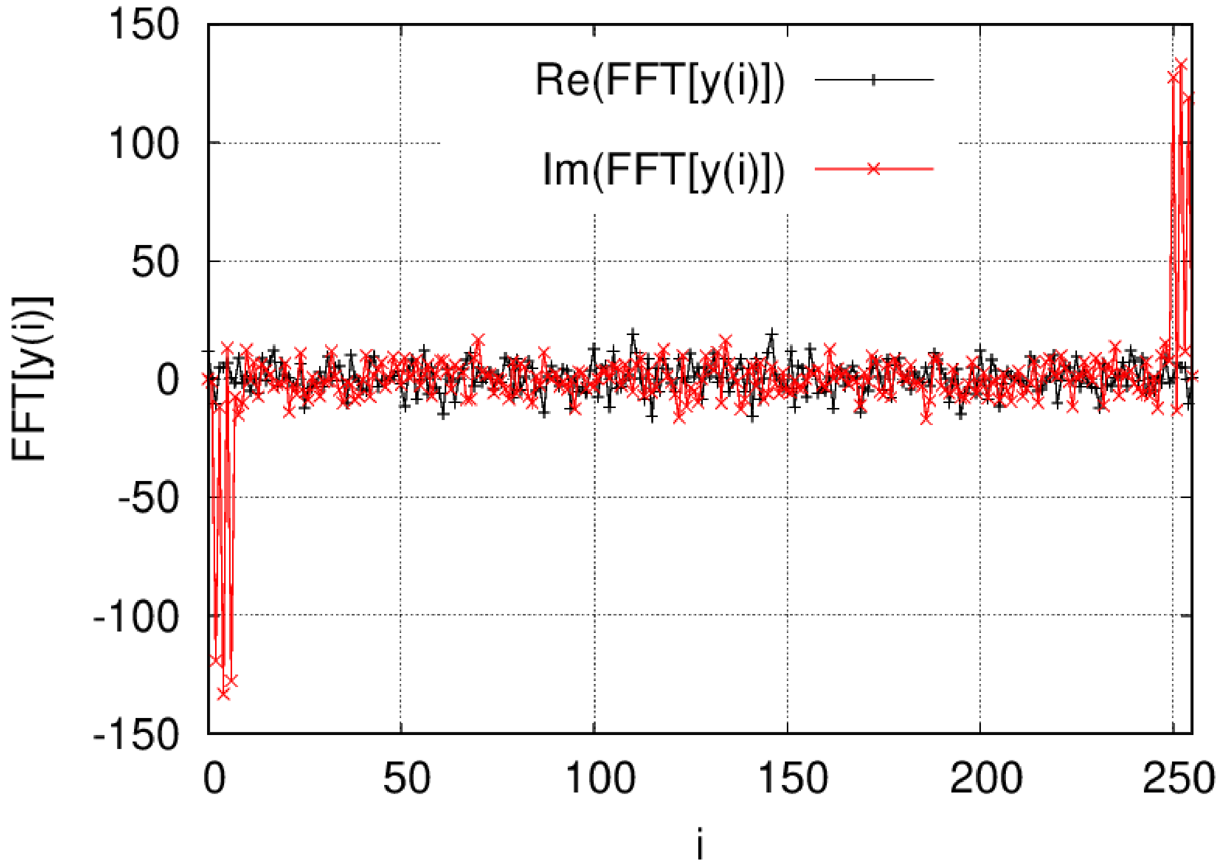
Dalej przeprowadziliśmy dyskryminację sygnału na poziomie max($\frac{|c_k|}{2}$), to znaczy że trzeba było znaleźć element o maksymalnym module, i wyzerować wszystkie elementy których moduł był mniejszy od $\frac{1}{2} c_{kmax}$.
Ostatnie co zrobiliśmy to wyznaczyliśmy transformatę odwrotną korzystając z funkcji biblioteki **GSL:**

gsl_fft_complex_radix2_backward (gsl_complex_packed_array y, size_t sride, size_t N),

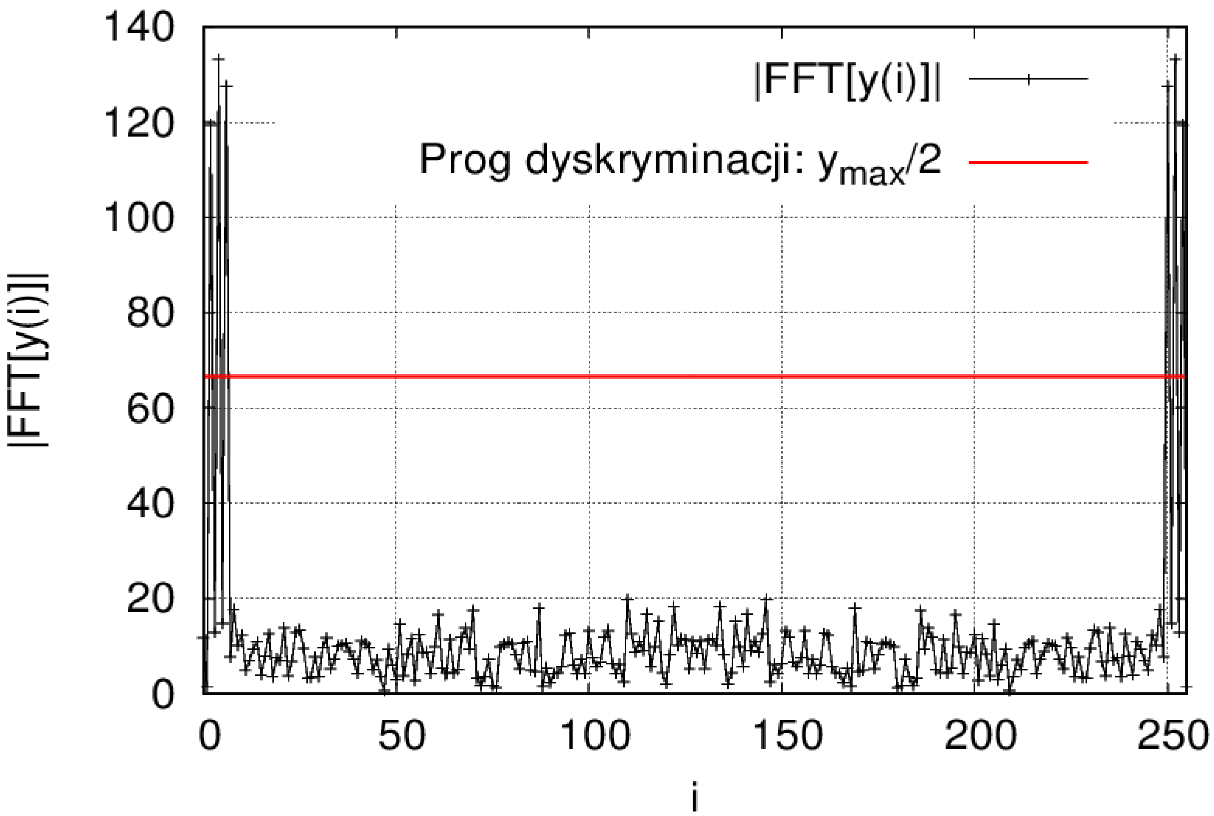
która przyjmuje argumenty analogiczne do funkcji **gsl_fft_complex_radix2_forward.**

3. Wyniki

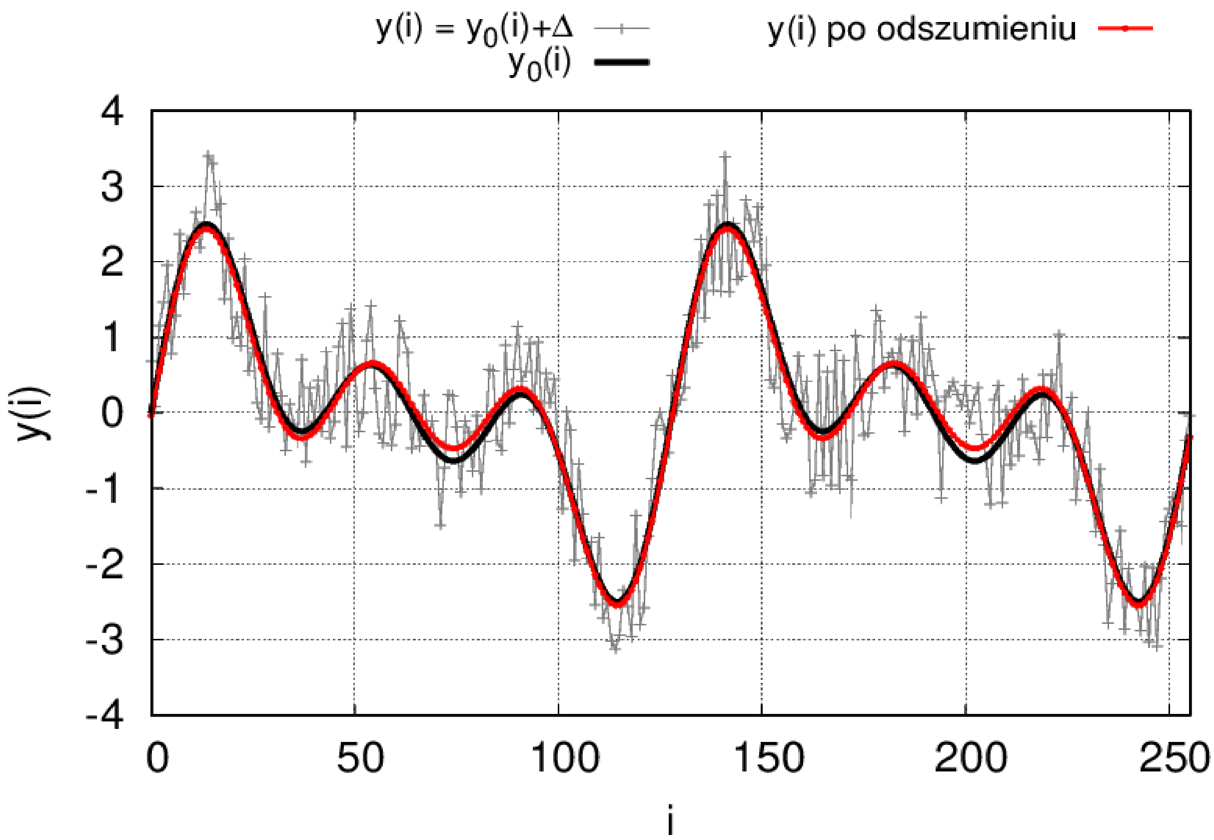
Aproksymację przeprowadzono kolejno dla $k = 8, 10, 12$. Dodatkowo dla k=8 przedstawiono wykres części rzeczywistej i urojonej transformaty oraz moduł liczby zespolonej w zależności od numeru iteracji.



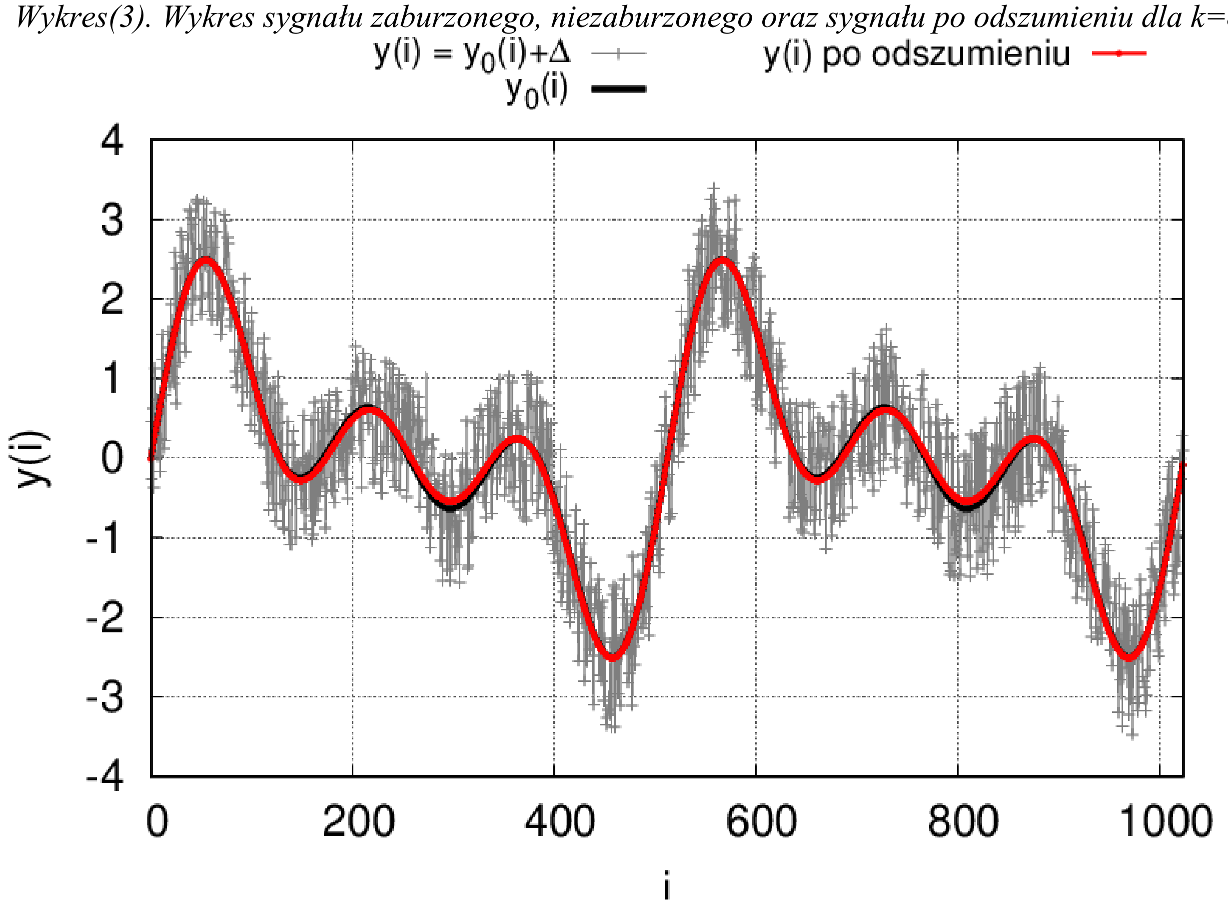
Wykres (1). Część rzeczywista i urojona transformaty. (k=8)



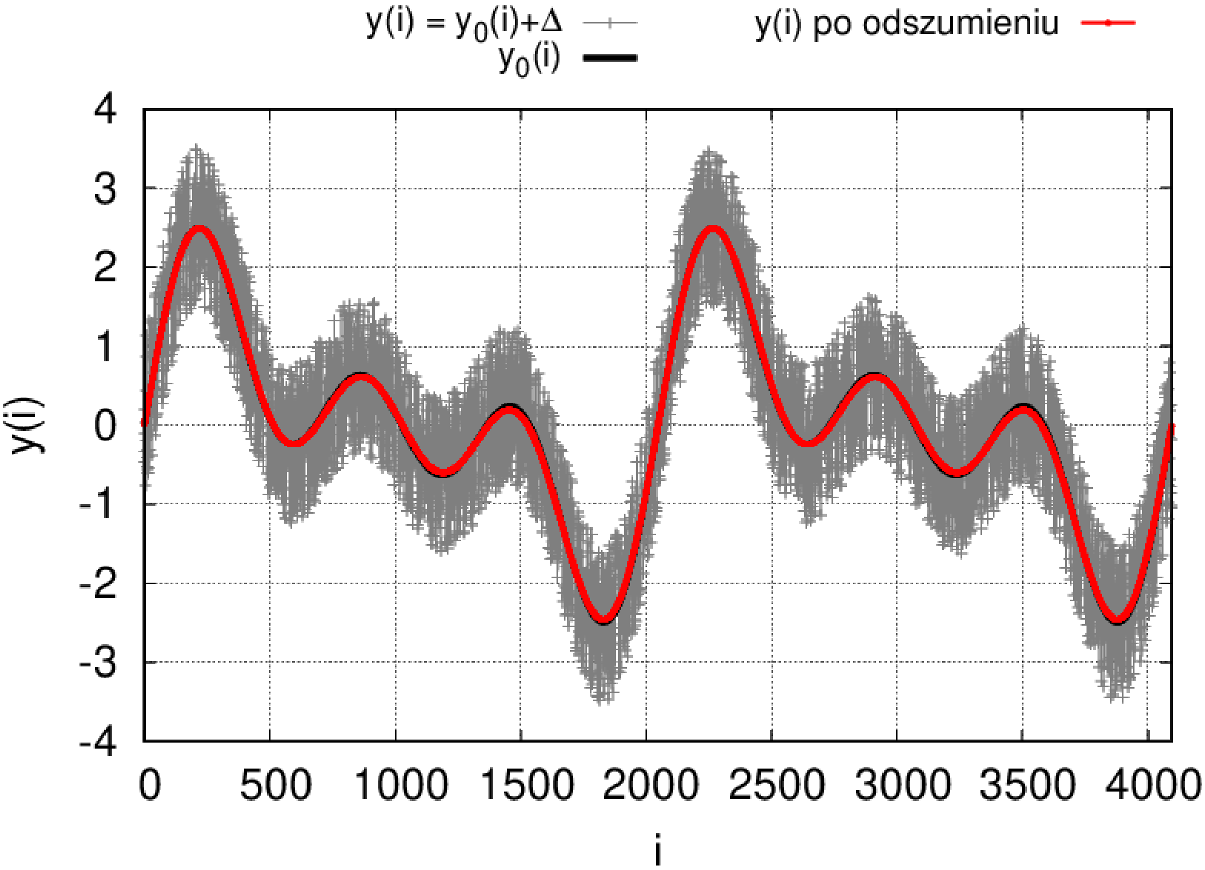
Wykres(2). wartości modułów współczynników transformaty(k=8)



Wykres(3). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=8



Wykres(4). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=10



Wykres(5). Wykres sygnału zaburzonego, niezaburzonego oraz sygnału po odszumieniu dla k=10

4. Wnioski

Dla każdej ilości próbek zaburzony sygnał został odszumiony. Z rysunków 3-5 można zauważyć że zwiększając liczbę próbek odszumienie bardziej się pokrywa z niezaburzonym sygnałem. Częstotliwość próbkowania ma zatem wpływ na dokładność procesu odszumiania.