***MOWNIT - Zestaw 4 Szybka transformata Fouriera***

*Opracował: Mateusz Woś*

Wszystkie zadania zostały zaimplementowane w języku Python.

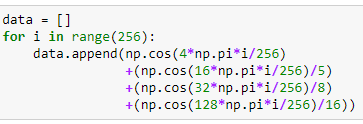
Korzystałem z bibliotek matplotlib do rysowania wykresów i biblioteki numpy, która zawiera wszystkie potrzebne funkcje z biblioteki gsl do wykonania sprawozdania.

1. *Proszę wygenerować 256-elementową tablicę wartości funkcji będącej sumą czterech cosinusów o różnych okresach i amplitudach, korzystając np. z wzoru: data[i]=cos(4\*Pi\*i/N)+cos(16\*Pi\*i/N)/5+cos(32\*Pi\*i/N)/8+cos(128\*Pi\*i/N)/16 gdzie N = 256. Narysować wykres tej funkcji korzystając z Gnuplota. Tę funkcję nazywamy sygnałem.*

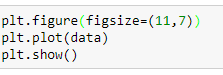
Import bibliotek:

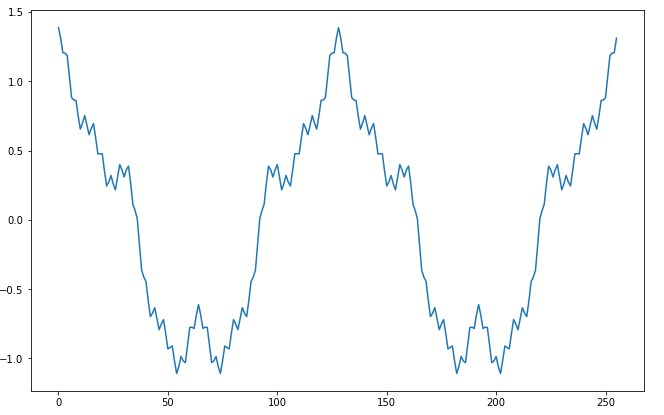


Wypełnienie tablicy danymi z zadania:



Wygenerowanie wykresu funkcji sygnału:



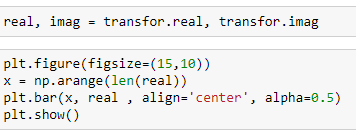


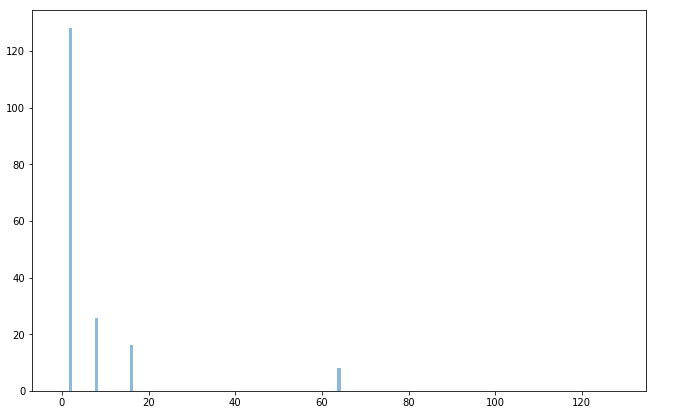
1. *Proszę użyć funkcji gsl\_fft\_real\_radix2\_transform do otrzymania transformaty Fouriera sygnału. Narysować wykres słupkowy tej transformaty. Czy na tym wykresie widać związek z częstotliwościami i amplitudami cosinusów wchodzących w skład funkcji? Ten wykres nazywamy widmem częstotliwości.*

Wykonanie transformaty Fouriera:



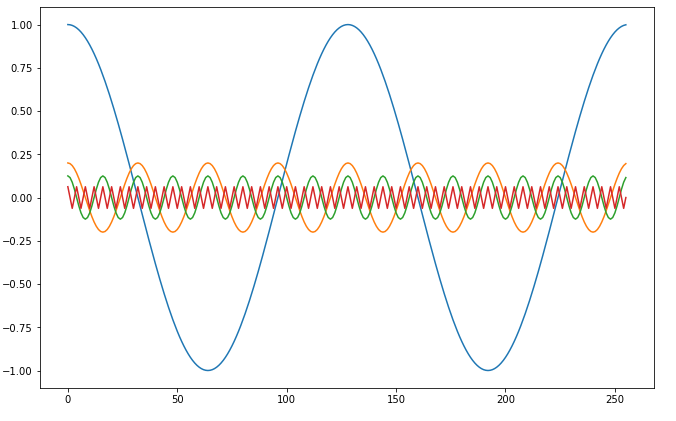
Stworzenie wykresu słupkowego:





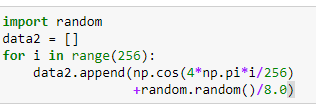
Na powyższych wykresie widać związek z częstotliwościami i amplitudami cosinusów wchodzących w skład funkcji. Argumenty w powyższym wykresie odpowiadają połowom częstotliwości cosinusów. Dla częstotliwości 4 mamy argument 2, dla 16 to 8, dla 32 to 16, a dla 128 to 64. Natomiast wartości słupków odpowiadają amplitudy. Dla nie zmienionej jest to 128, kolejna to podzielona przez 5, następne to podzielona przez 8 – 8-krotnie mniejsza i ostatni słupek odpowiada tej podzielone przez 16 – jest 16krotnie mniejsza od pierwszej wartości.

Wykres cosinusów składających się na sygnał:

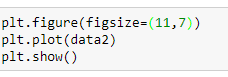


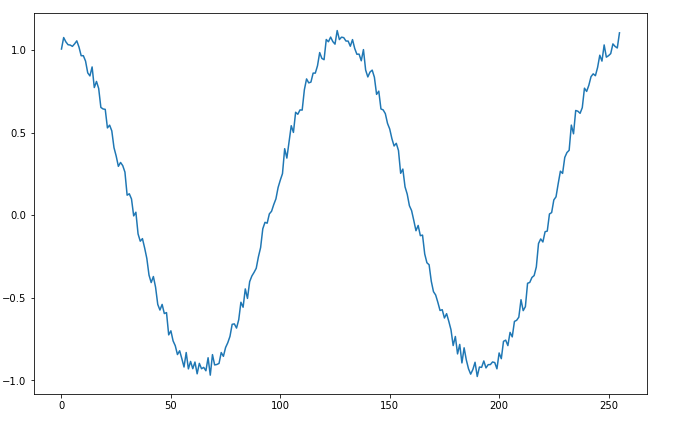
1. *Teraz wypełniamy tablicę wartościami funkcji cosinus zaburzonej niewielkim "szumem", np. korzystając z wzoru: data[i]=cos(4\*Pi\*i/N)+((float)rand())/RAND\_MAX/8.0 Proszę narysować wykres tej funkcji.*

Wypełnienie tablicy danymi:



Wykres:

**

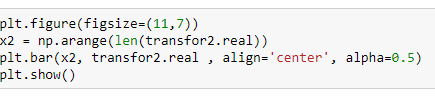
**

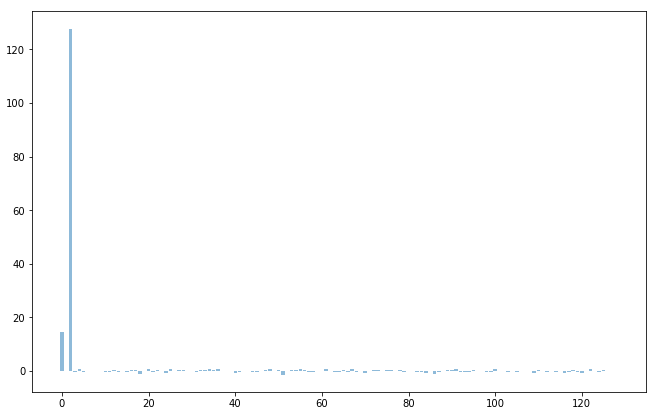
1. *Narysować wykres transformaty Fouriera tego sygnału (jak w punkcie 2).*

Wykonanie transformaty Fouriera:



Stworzenie wykresu słupkowego:

**

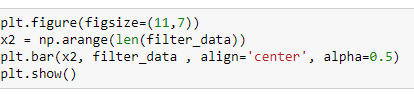
**

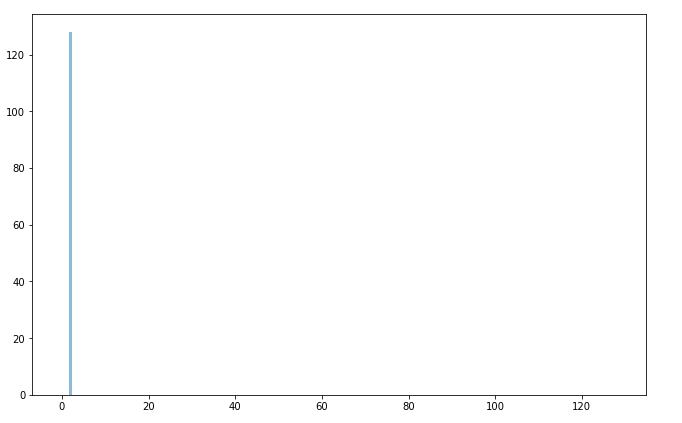
1. *Po transformacie wyzerować w widmie wszystkie elementy, których wartość bezwzględna jest mniejsza niż 50. W ten sposób usuwamy "szumy" z sygnału.*

Usuwanie szumów:



Wykres:



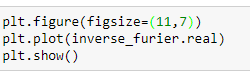


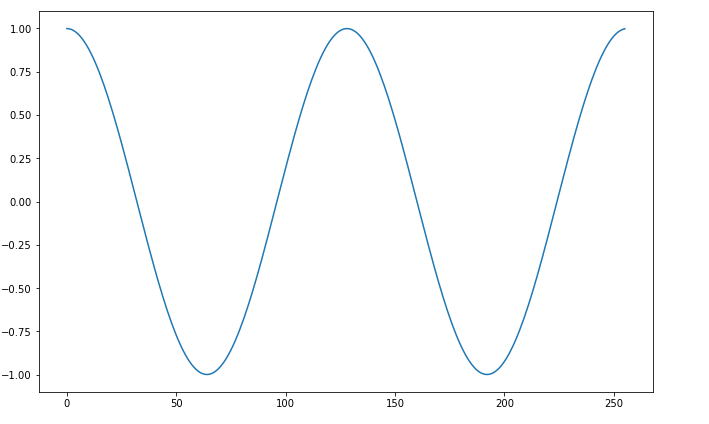
1. *Użyć funkcji gsl\_fft\_halfcomplex\_radix2\_inverse do przeprowadzenia odwrotnej transformaty. Narysować wykres otrzymanej funkcji. Czym różni się ona od wyjściowego sygnału?*

Odwrotna transformata Fouriera:



Wykres:





Otrzymałem wersję bez szumów funkcji cosinus. Funkcja już nie zawiera w sobie „zanieczyszczeń” z poprzedniej wersji.