TECHNIKI PROGRAMOWANIA – projekt 3

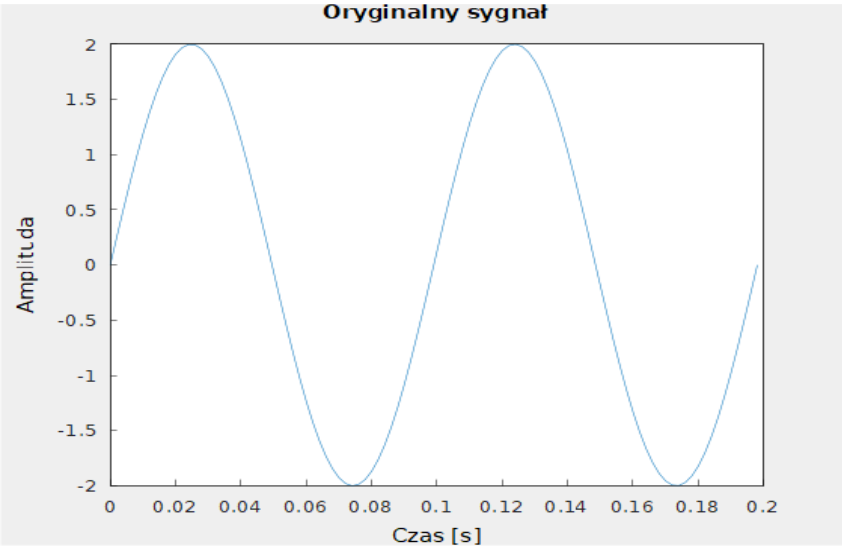
Michał Rachimow 203333
Antoni Rachwał 203718

Wykorzystane biblioteki:
-matplotplusplus (c++)
-pybind11 (c++)
-vector (c++)
-cmath (c++)
-complex (c++)

Wizualizacja wykresów 1D dla nadanych parametrów

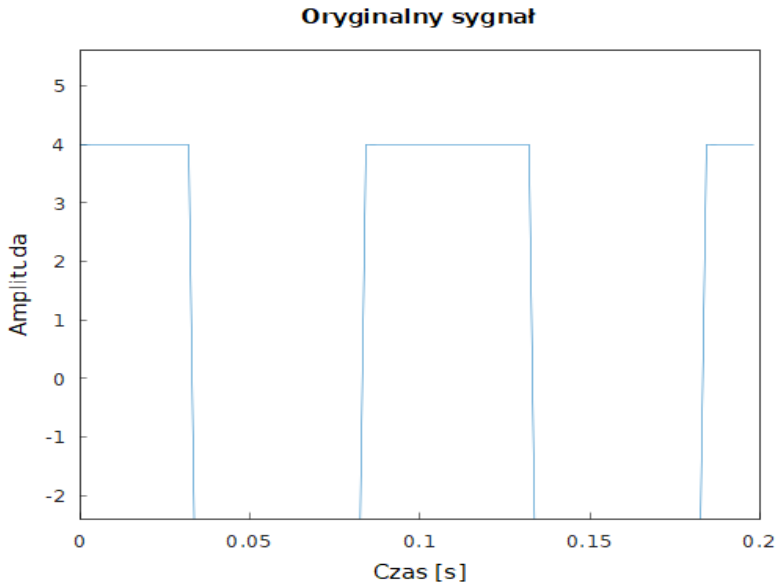
```
import example

frequency = float(input("Podaj częstotliwość (Hz): "))
amplitude = float(input("Podaj amplitudę: "))
phase = float(input("Podaj fazę (w radianach): "))
samples = int(input("Podaj liczbę próbek: "))

signal = example.generate_sine(frequency, amplitude, phase, samples)

example.plot_signal(signal, "Oryginalny sygnał")
```

Wykres sinusa dla:
frequency =10.0
amplitude =2.0
phase =0.0
samples =1000



signal = example.generate_square(frequency, amplitude, phase, samples)

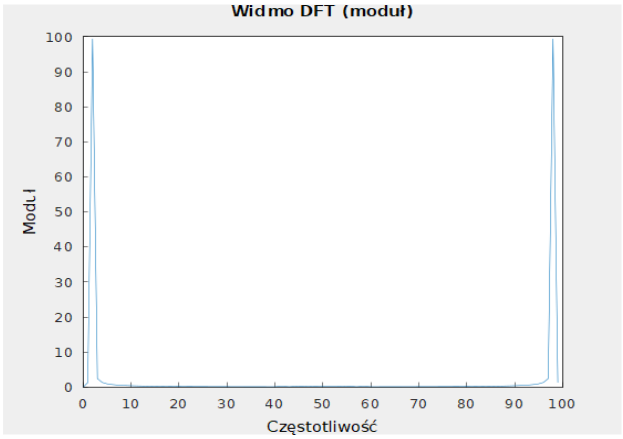
Wykres o tych samych wartościach dla przebiegu prostokątnego:

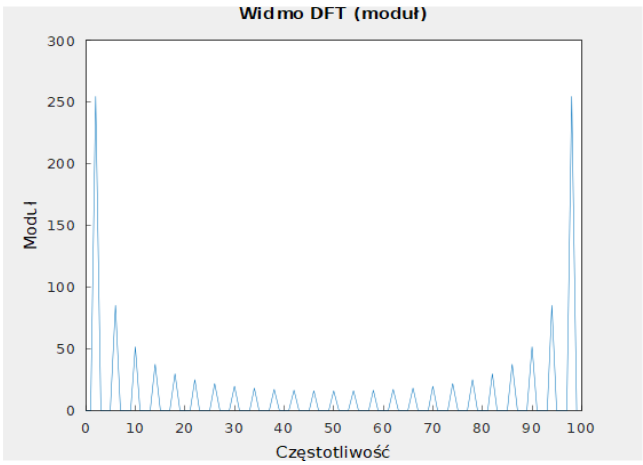


DFT i IDFT

dft_result = example.compute_dft(signal)
example.plot_magnitude(dft_result, "Widmo DFT (moduł)")

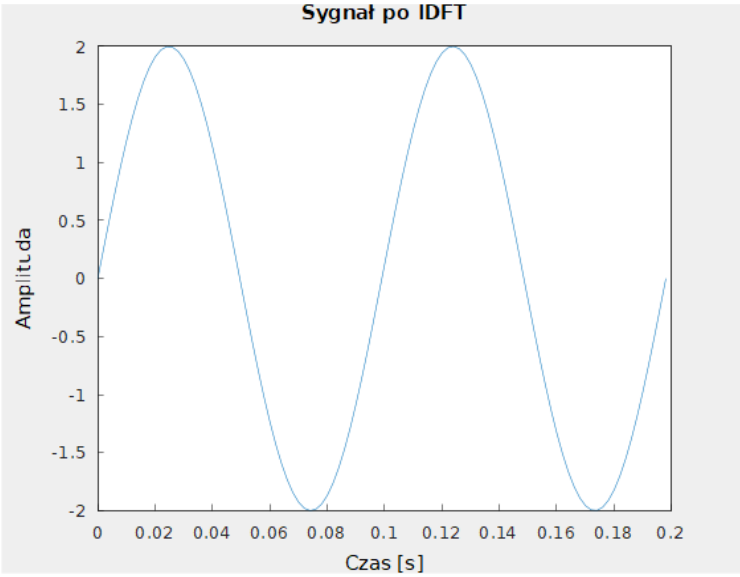
Widma DFT wygenerowanych wyżej sygnałów:

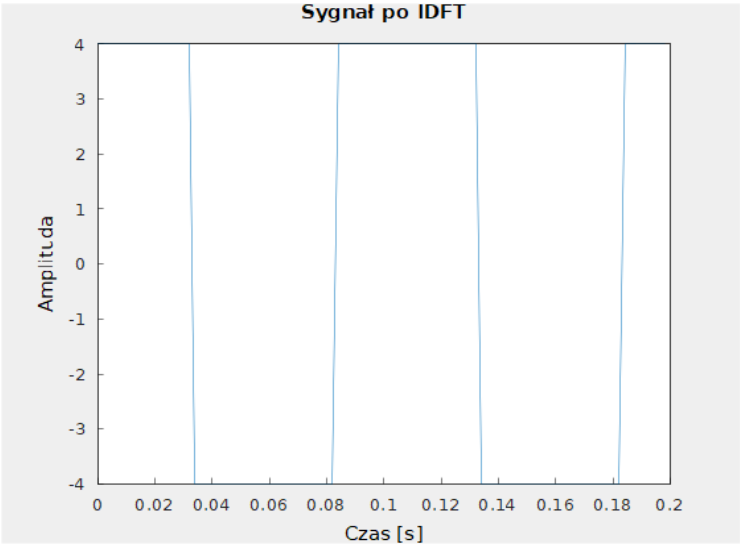




Aby wykonać transformację odwrotną należy zaimplementować:

reconstructed = example.compute_idft(dft_result)
example.plot_signal(reconstructed, "Sygnał po IDFT")

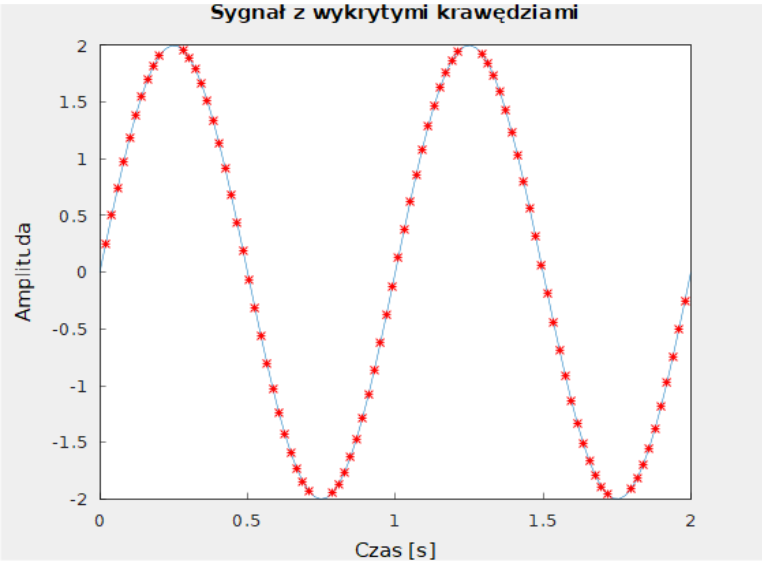


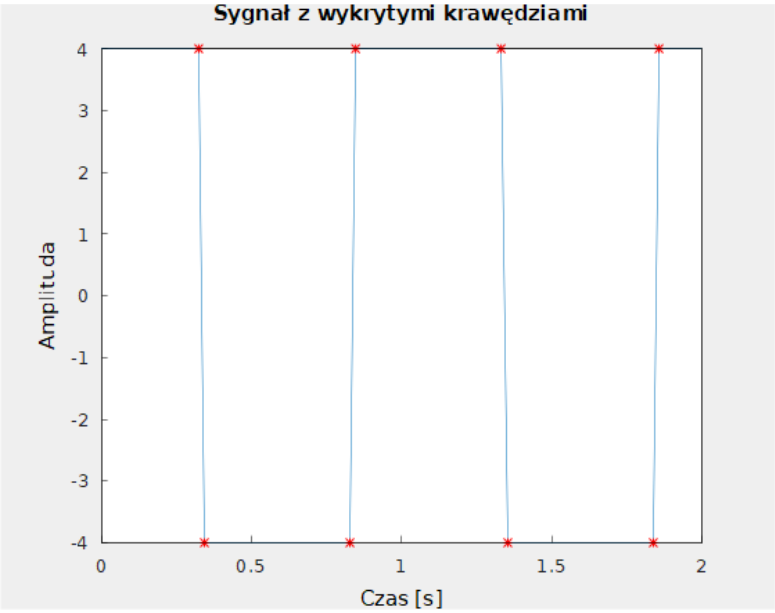


Wykrywanie krawędzi:

edges = example.detect_edges(signal, threshold=0.1)

example.plot_signal_with_edges(signal, edges, "Sygnał z wykrytymi krawędziami")





Sygnał piłokształtny: