Sprawozdanie nr. 3

Szeregi Fouriera

analiza

Hubert Król

2. Wstęp

a) Cel ćwiczenia

W tym sprawozdaniu obliczę transformaty Fouriera algorytmem (DFT i FFT) oraz odwrotne transformaty Fouriera (IDFT i IFFT) dla jednego i dwóch wymiarów. Przeprowadzę analizę widma i usunę zakłócenia

b) Niezbędne podstawy teoretyczne

DFT

Obliczenie DFT sygnału sprowadza się do wyznaczenia wartości liczbowych sumy

$$\boldsymbol{X(k) = \sum_{n=0}^{N-1} x\,(n)\, exp\left(\frac{-2i \cdot \Pi kn}{N}\right)}$$

Wzór 1

gdzie k=0, 1,...,N−1 dla przekształcenia prostego (czas → częstotliwość) oraz sumy.

$$\boldsymbol{x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X\,(k)\, exp\left(\frac{2i\Pi kn}{N}\right)}$$

Wzór 2

FFT

Mamy N próbek sygnału dyskretnego (N musi być potęgą dwójki). Dzielimy ten ciąg na dwa ciągi: $xp(n)$, $xn(n)$, składające się z próbek sygnału $x(n)$ o indeksach odpowiednio parzystych (0, 2, 4, ...) i nieparzystych (1, 3, 5, ...). Obliczamy transformaty Fouriera tych ciągów: $Xp(n)$, $Xn(n)$, wywołując rekurencyjnie procedurę obliczania FFT. Następnie składamy transformatę całego ciągu próbek, zgodnie z zależnością:

$$X(k)=Xp(k)+WNkXn(k)$$
$$X(k+N/2) =Xp(k)-WNkXn(k)$$
$$WNk=e-2\pi jk/N$$

Zależność 1

dla $0\leq k<N/2$. Tak więc obliczanie FFT polega na kolejnym dzieleniu próbek na ciągi parzyste i nieparzyste, aż do otrzymania dwupunktowych ciągów, dla których $X(0)=x(0)+x(1)$,
$X(1)=x(0)-x(1)$. Następnie transformaty ciągów parzystych i nieparzystych są składane w całość.

3. Wymagania techniczne

Algorytm został opracowany na

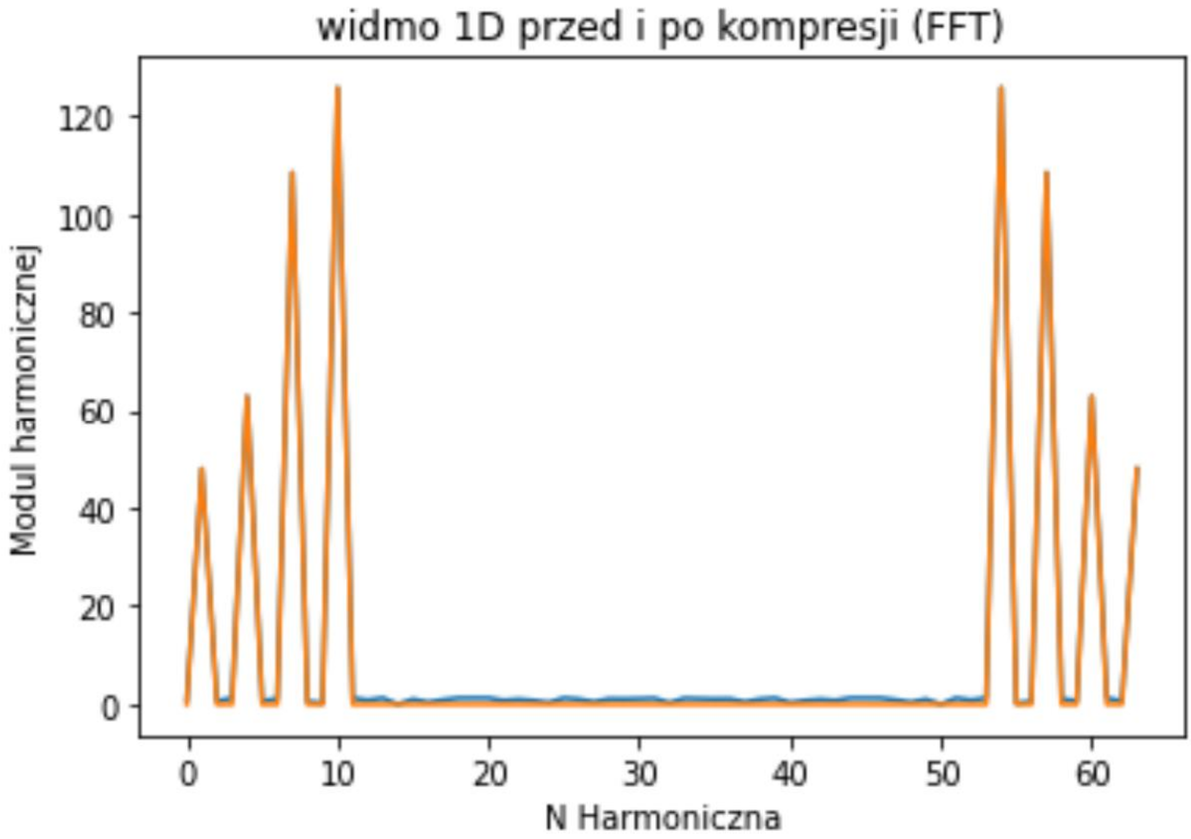
- Anaconda Navigator (anaconda 3) Spyder (Python3.9)
- Windows 10

Biblioteki:

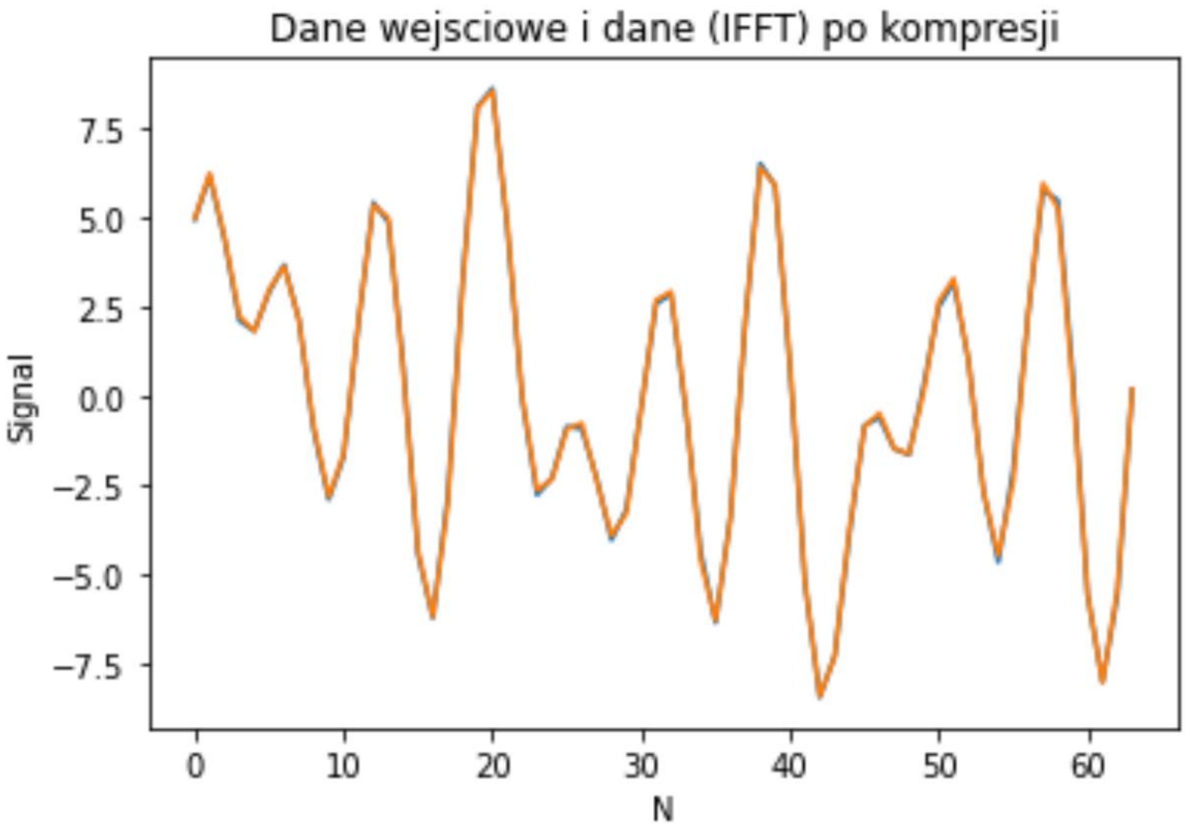
- Numpy
- Pandas
- Warnings
- Cmath
- Matplotlib
- Sys

4. Przedstawienie wyników

4.1 Widmo sygnału FFT dla jednego wymiaru:



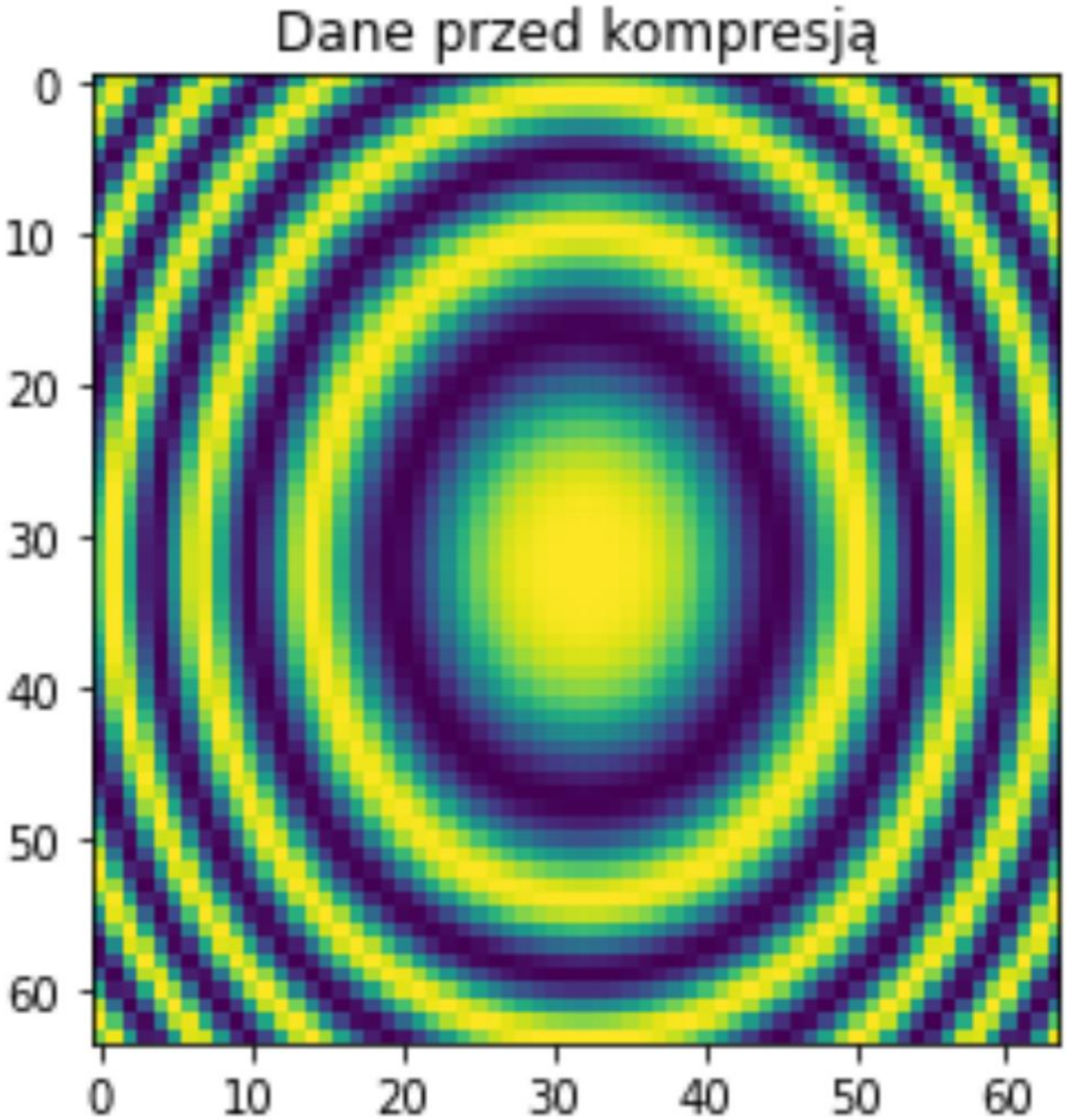
Obraz 1. Widmo sygnału dla 1 przed i po kompresji dla jednego wymiaru dla pliku Dane_03_a



Obraz 2. Dane wejściowe i dane po zastosowaniu IFFT po kompresji dla pliku Dane_03_a

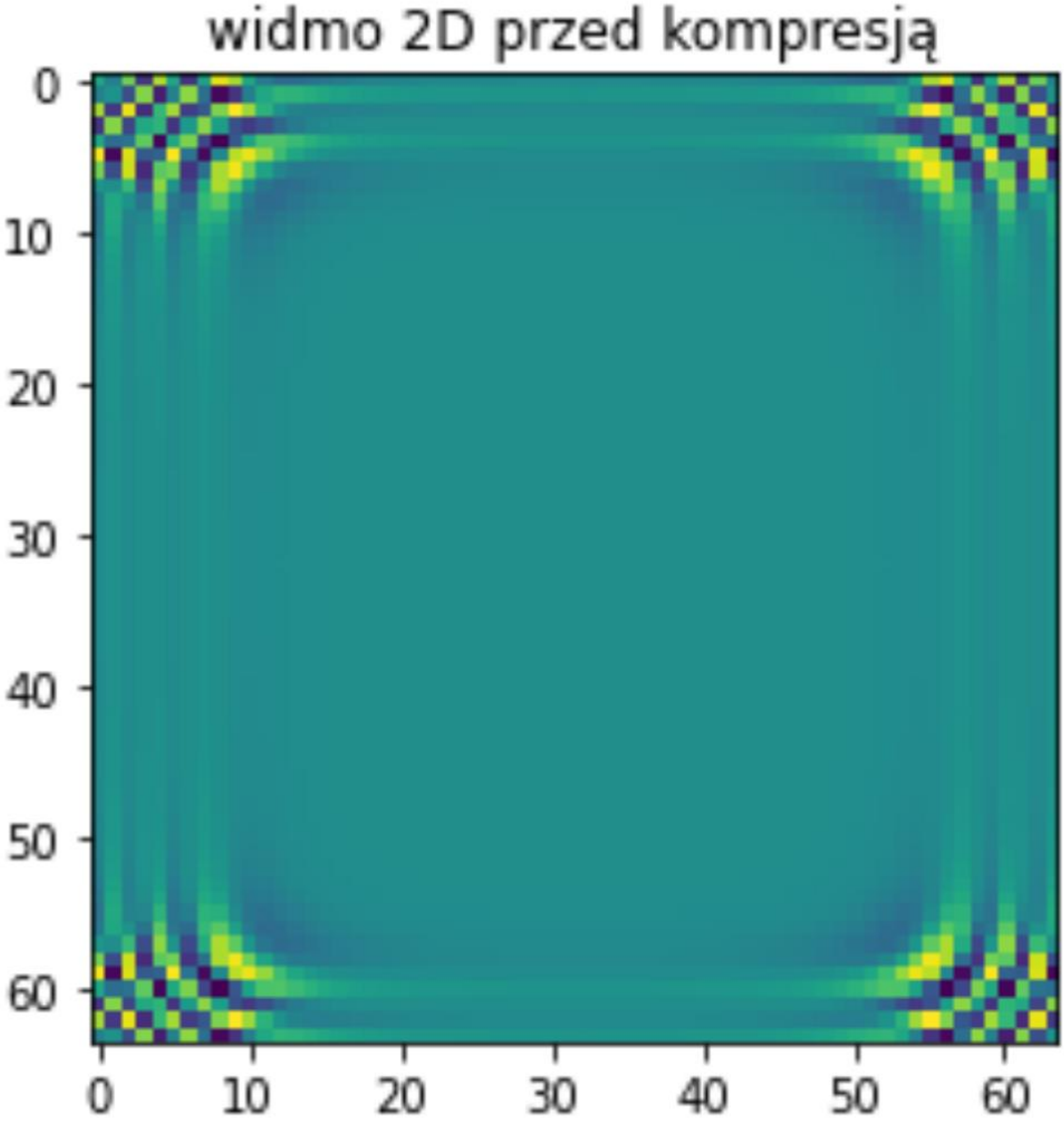
Kompresji dokonałem na podstawie średniej modułu wszystkich danych. Jeżeli moduł poszczególnego współczynnika był mniejszy od tej wartości, to był zerowany.

4.2 Widmo sygnału FFT dla dwóch wymiarów:



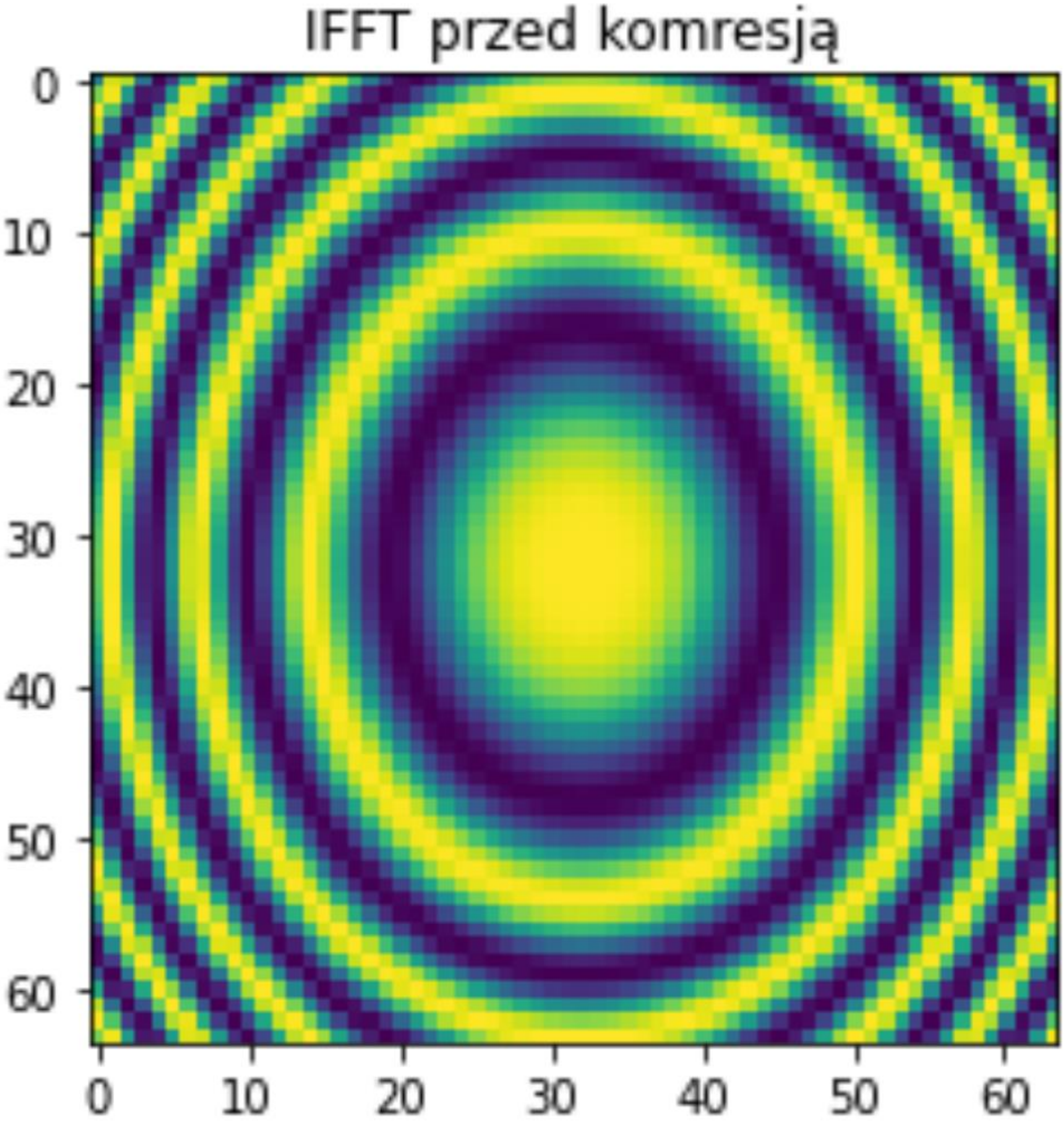
Obraz 1. Dane wejściowe dwuwymiarowe przed kompresją dla pliku Dane2_03.

Zastosowanie FFT na widmie przed kompresją

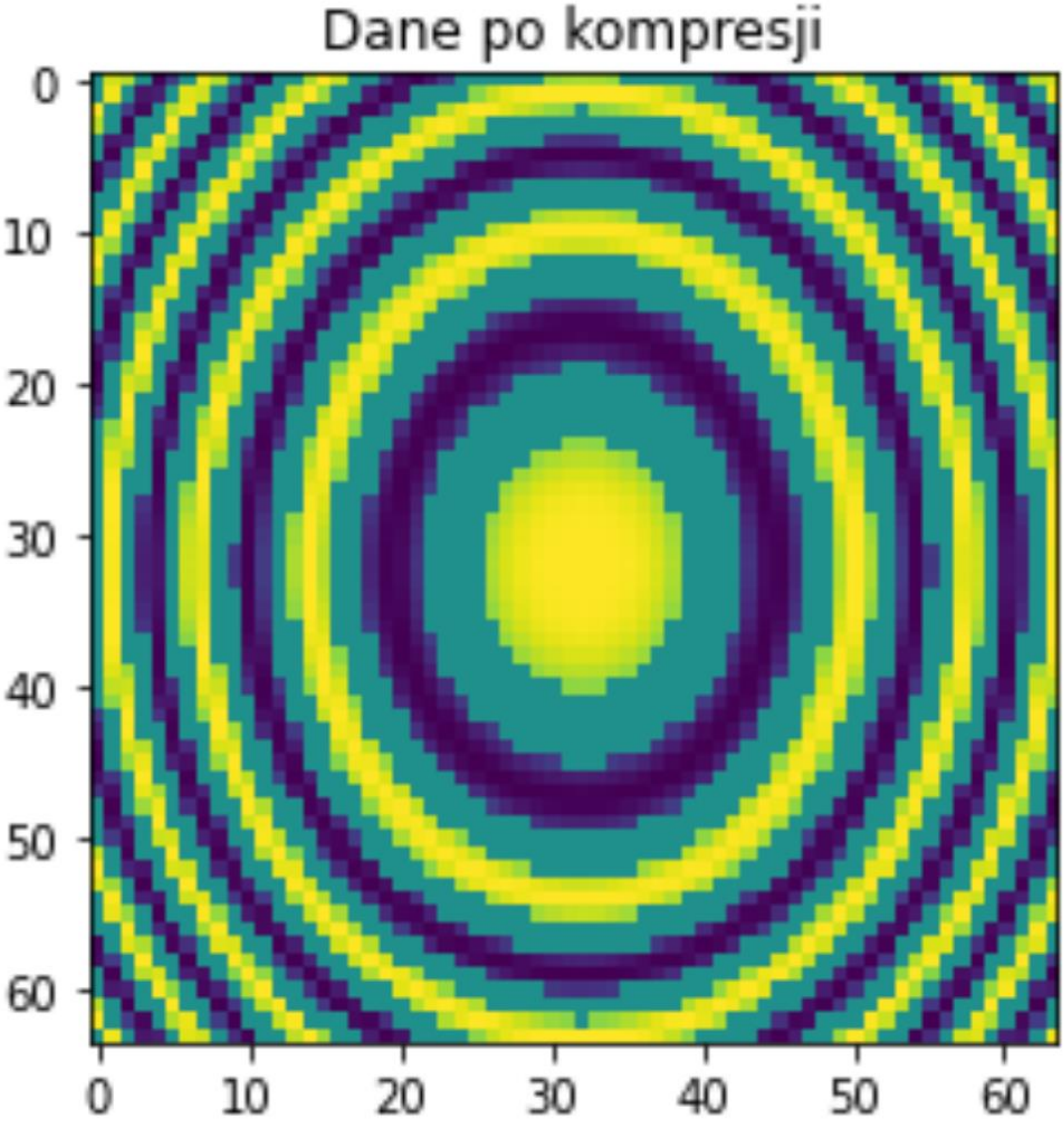


Obraz 2. Widmo 2wymiarowe przed kompresją dla pliku Dane2_03.

Zastosowanie IFFT na widmie przed kompresją

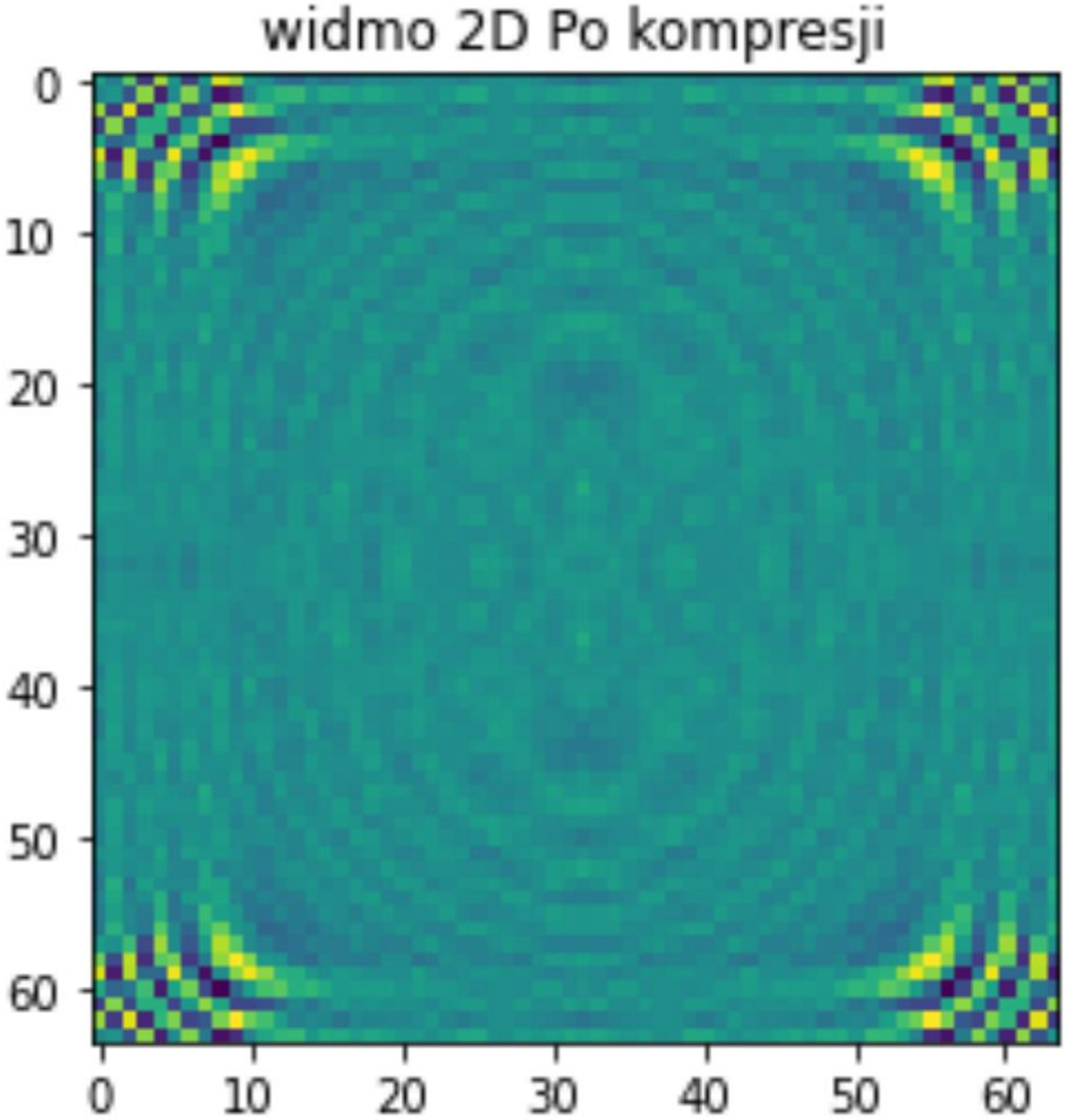


Obraz 3. IFFT przed kompresją dla pliku Dane2_03.



Obraz 3. Dane wejściowe dwuwymiarowe przed kompresji dla pliku Dane2_03.

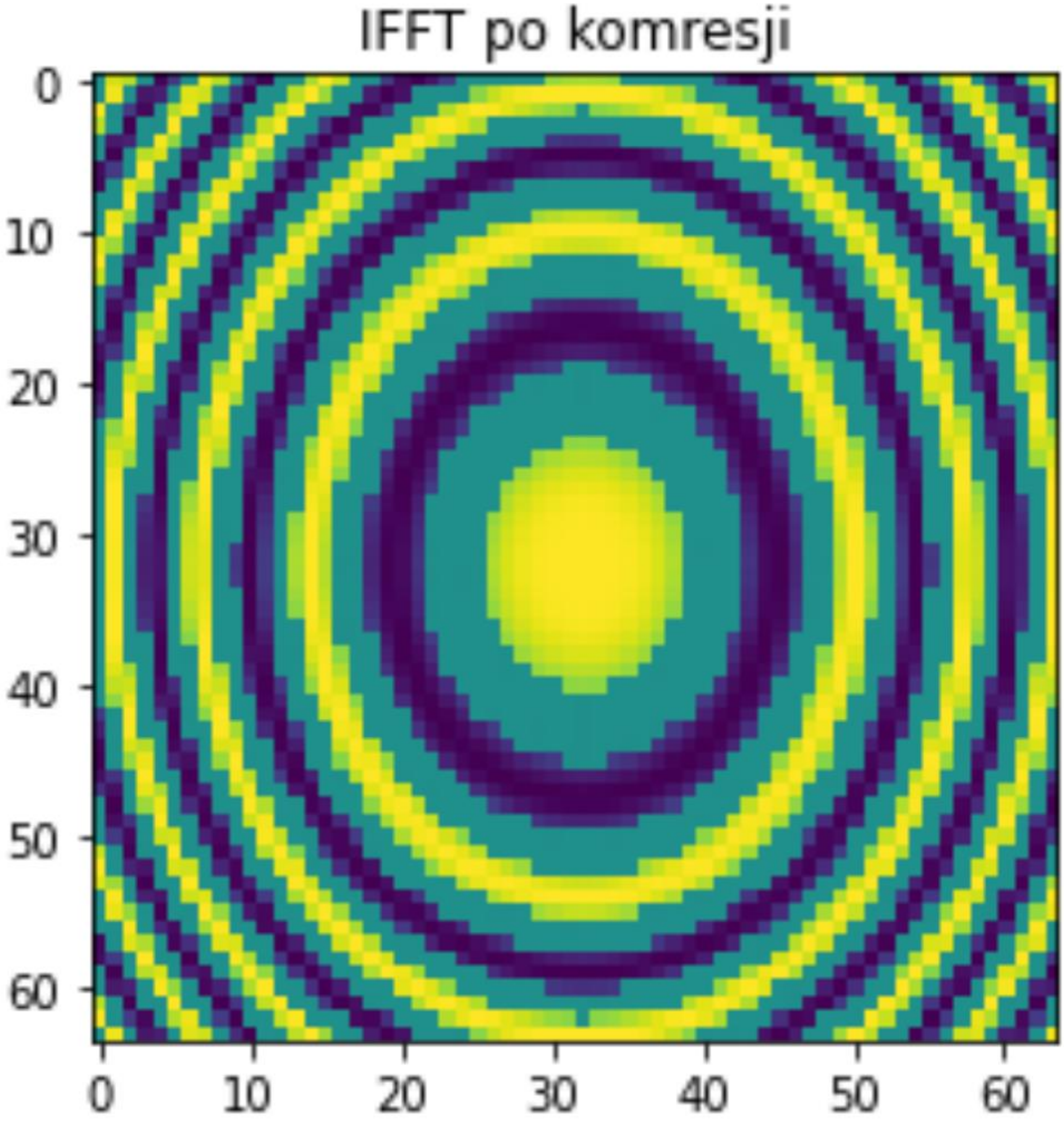
Zastosowanie FFT na danych wejściowych po kompresji



Obraz 4. Widmo dwuwymiarowe po kompresji dla pliku Dane2_03.

Po kompresji obraz widma zrobił się ostrzejszy

Zastosowanie IFFT na widmie po kompresji



Obraz 6. IFFT po kompresji dla pliku Dane2_03.

Po kompresji obraz IFFT zrobił się ostrzejszy zmieniły się też granice linii

5. Wnioski

Dokonując analizy, z której dokonałem w punkcie 4 wysuwam wniosek, że dane przed i po kompresji dla jednego wymiaru nie mają wielkiej różnicy jednak dla większej liczby wymiarów różnica jest zauważalna obraz pom kompresji jest dużo ostrzejszy

Przyczyną tej różnicy jest funkcja determinująca czy dany współczynnik ma znaczenie czy też nie. Porównanie do średniej ma istotny wpływ na wynik końcowy, gdyż niektóre linie mają zbyt małe znaczenie ze względu na ich niski moduł.

6.Bibliografia

- https://sound.eti.pg.gda.pl/~greg/dsp/01-AnalizaWidmowa.html

- https://www.fuw.edu.pl/~prozanski/AS/?ft-calc/dft