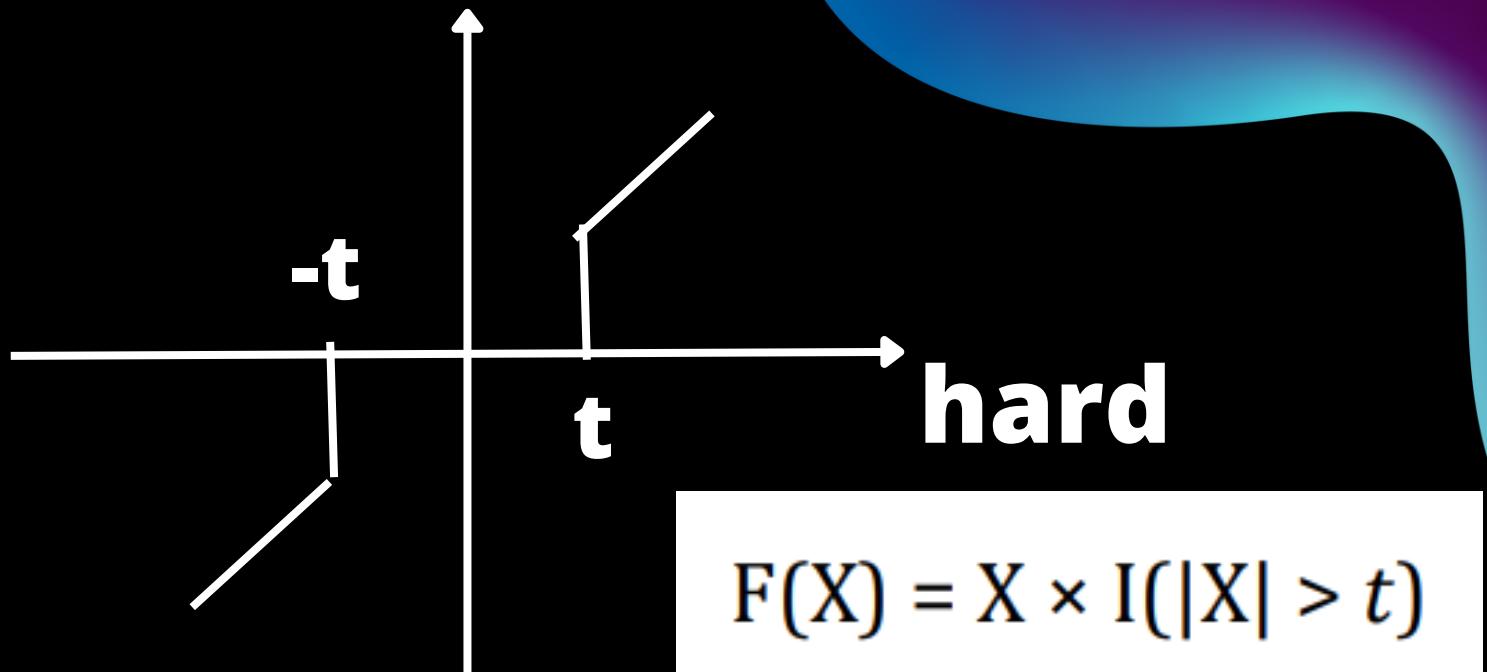
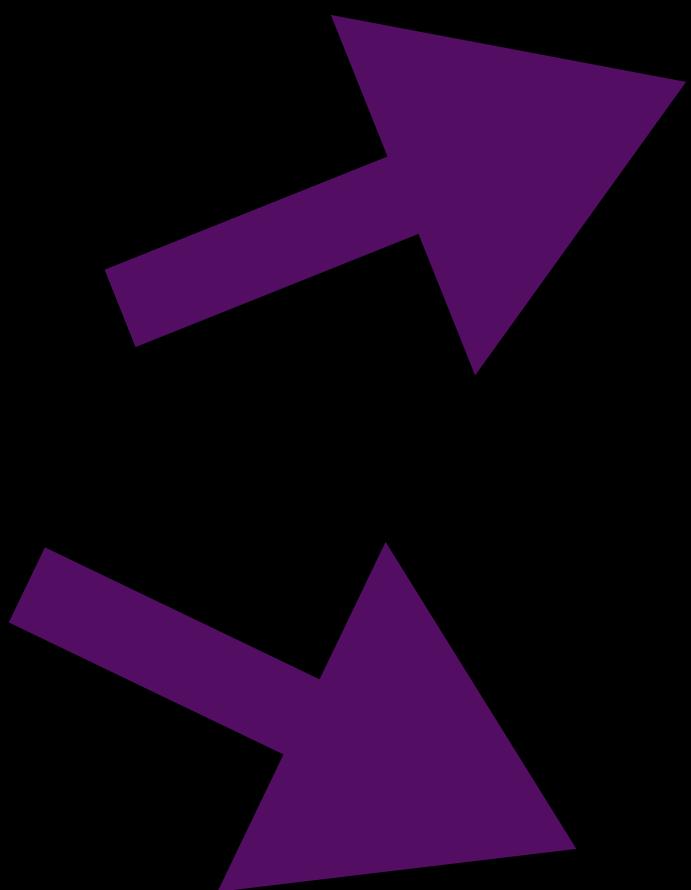
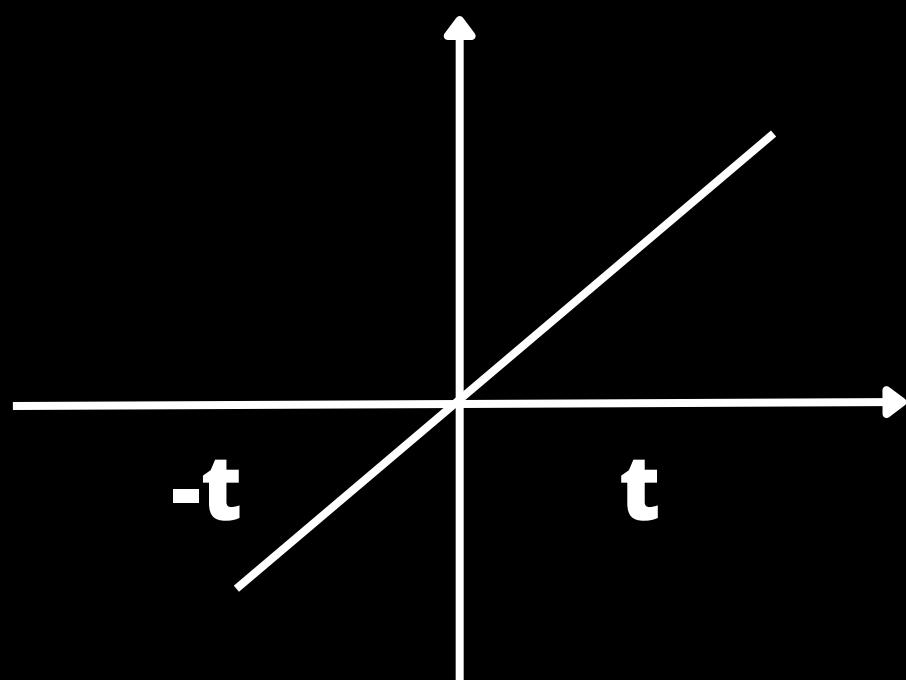




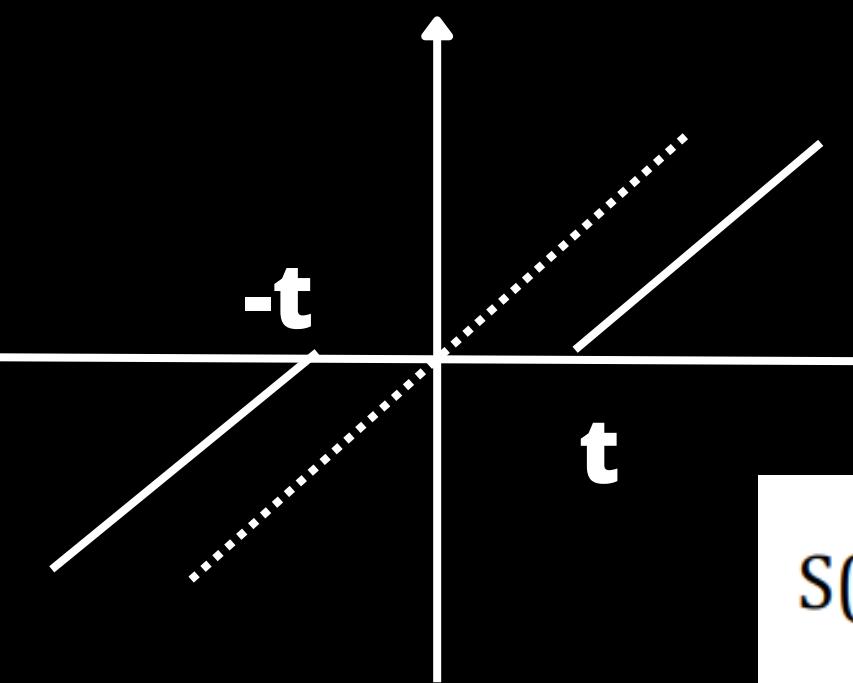
ODSZUMIANIE
SYGNAŁU



Soft/hard thresholding

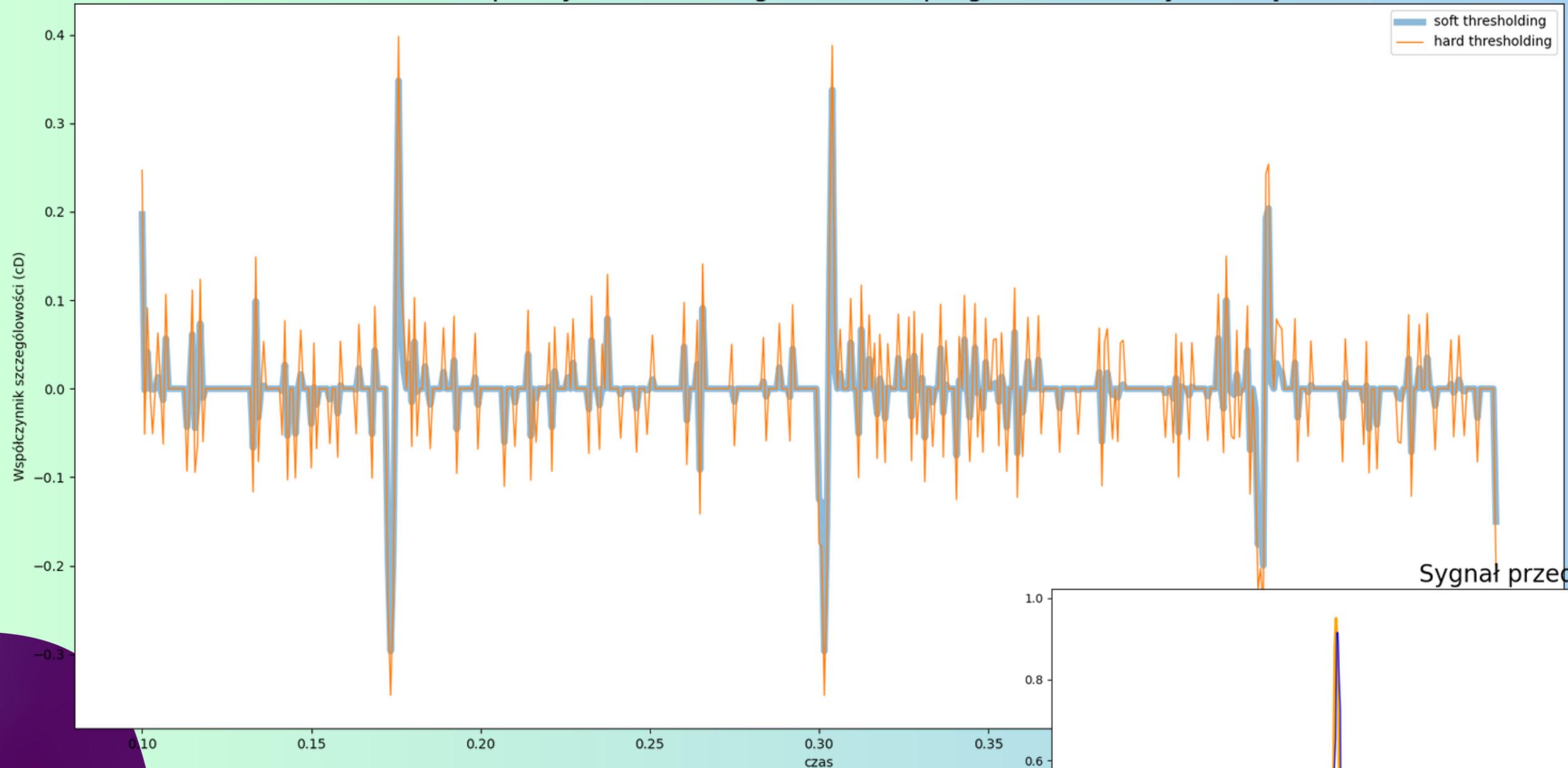


$$F(X) = X \times I(|X| > t)$$

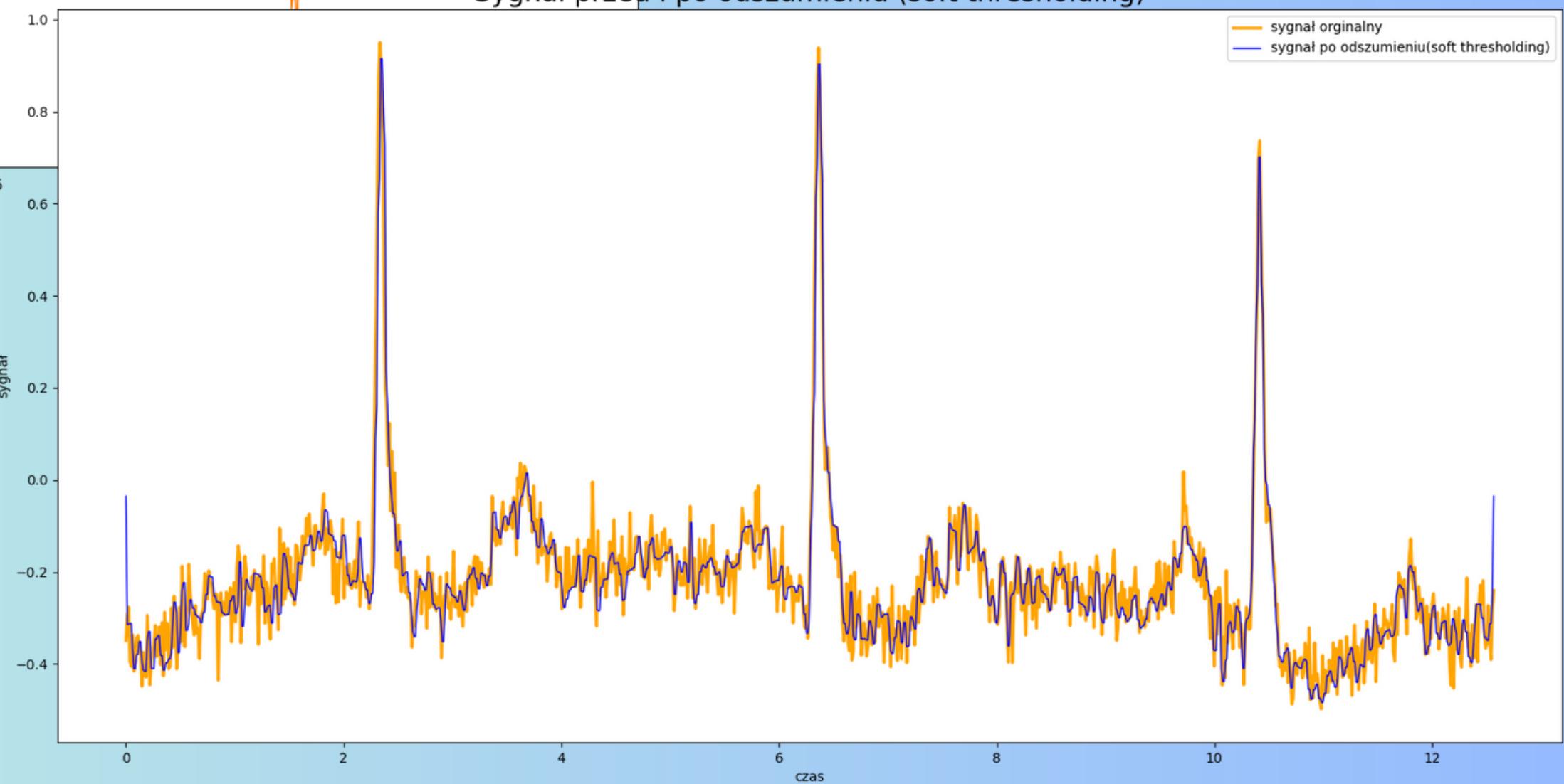


$$S(x) = \text{sign}(x)(|x| - t) I(|x| > t)$$

Porównanie współczynników szczegółowości w progowaniu twardym i miękkim



Sygnal przed i po odszumieniu (soft thresholding)

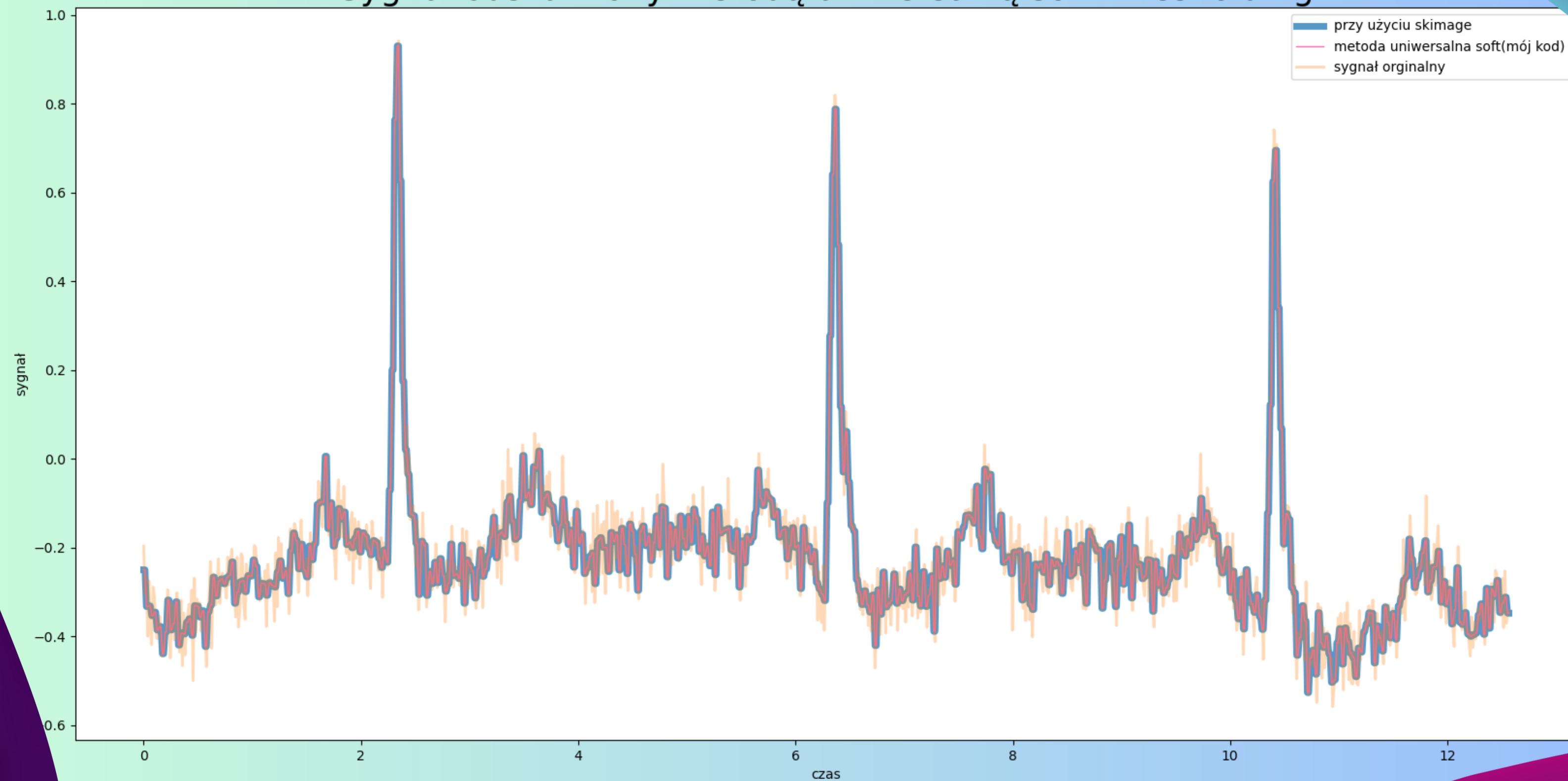


Metoda uniwersalna

$$T = \sigma \sqrt{2 \log(M)}$$

$$\sigma = \frac{\text{Median } |cD|}{0.6745},$$

Sygnal odszumiony metodą uniwersalną soft thresholding



Metoda Bayes-Shrinka

$$\hat{x} = \delta(x)y + (1 - \delta(x))\mu$$

$$\delta(x) = \frac{\sigma^2}{\sigma^2 + \tau^2}$$

```
import numpy as np
import pywt
import matplotlib.pyplot as plt

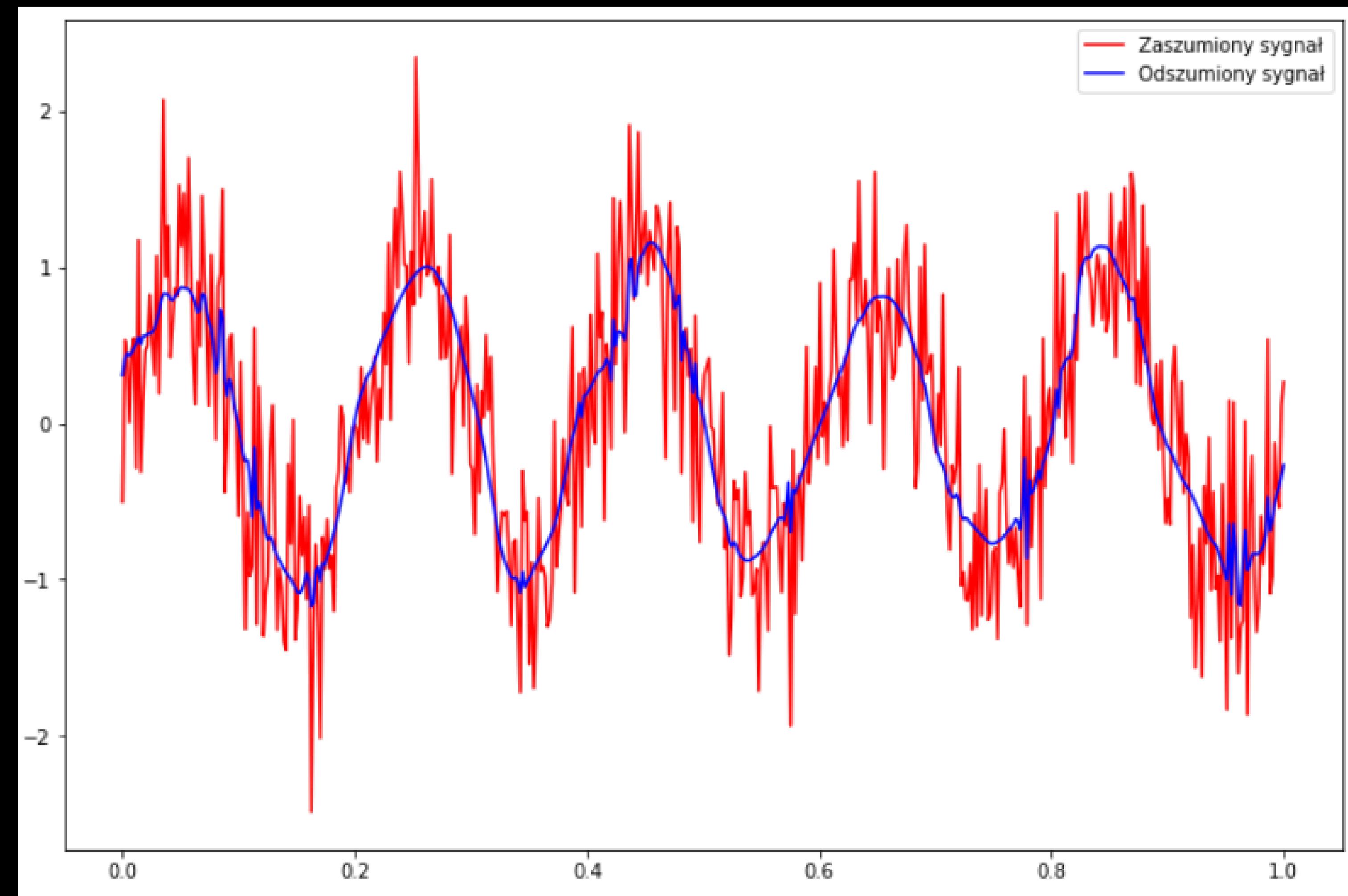
# Generowanie zaszumionego sygnału
x = np.linspace(0, 1, num=512)
signal = np.sin(10*np.pi*x) + np.random.normal(0, 0.5, size=512)

# Wyznaczenie współczynnika Bayesa dla każdego punktu sygnału
sigma = np.std(signal)
tau = 0.1 * sigma # stała Shrinkage
delta = sigma**2 / (sigma**2 + tau**2)

# Odszumianie sygnału
coeffs = pywt.wavedec(signal, 'db4', level=5)
for i in range(1, len(coeffs)):
    coeffs[i] = pywt.threshold(coeffs[i], delta*sigma)

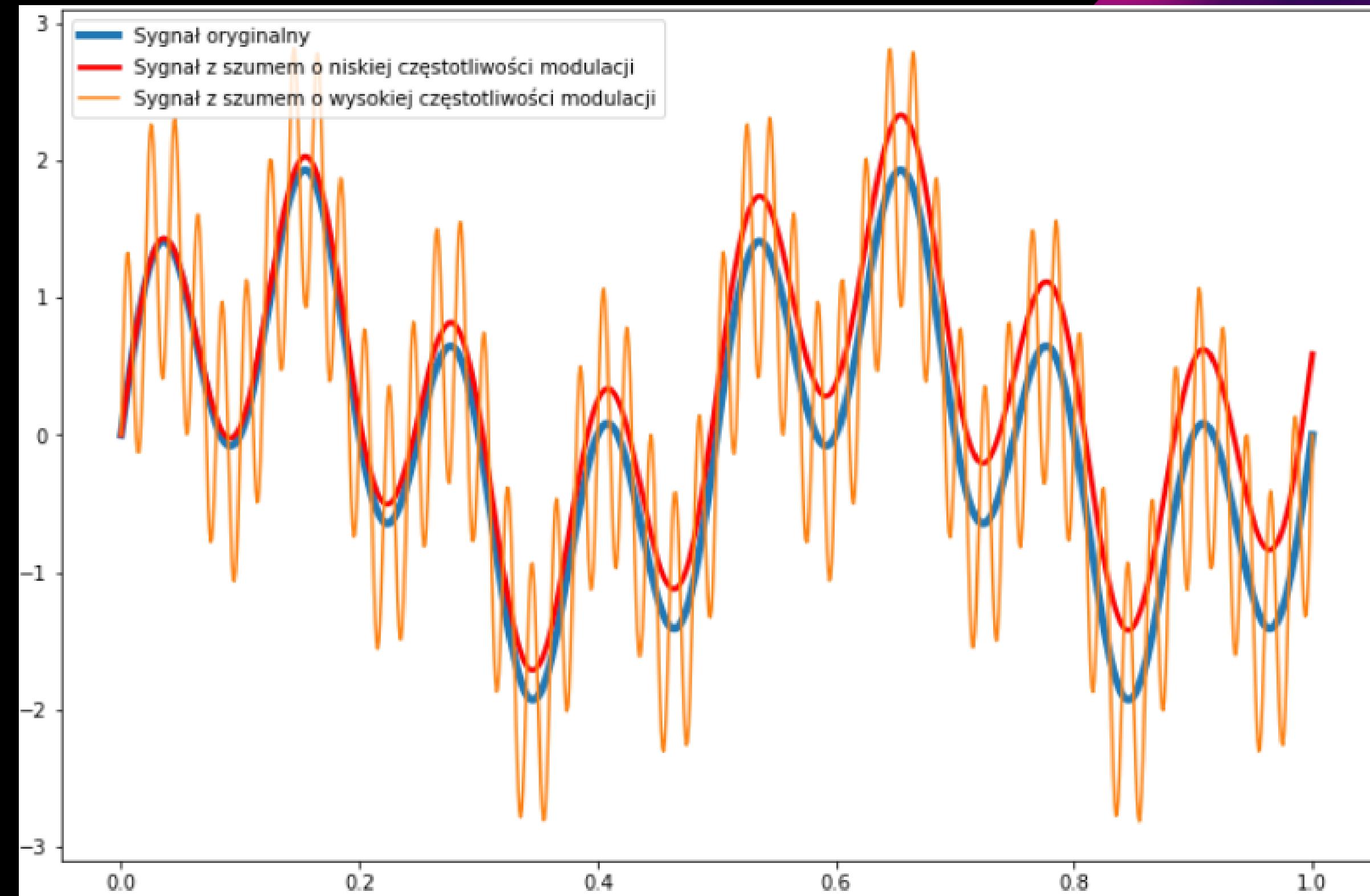
denoised_signal = pywt.waverec(coeffs, 'db4')

# Wykres
plt.plot(x, signal, 'r', label='Zaszumiony sygnał')
plt.plot(x, denoised_signal, 'b', label='Odszumiony sygnał')
plt.legend()
plt.show()
```



Zastosowanie metody Bayes-Shrinka

**niska częstotliwość
i złożone struktury**



Jak wydobyć śpiew ptaków za pomocą falki Morleta?



