Metody numeryczne

Laboratorium 11

Odszumianie sygnału przy użyciu FFT – splot funkcji

Anastasiya Hradouskaya

30.05.2021

Cel ćwiczenia

Zapoznanie się z metodą odszumiania sygnału przy użyciu FFT (Szybkiej Transformaty Fouriera).

1. Opis problemu

Za pomocą metody FFT znaleźć wykres funkcji odszumionego sygnału.

Sygnał bez zakłóceń wyraża się następującą funkcją:

$$f_0(t) = \sin(1 \cdot \omega t) + \sin(2 \cdot \omega t) + \sin(3 \cdot \omega t)$$

Na początek sygnał zaszumiamy:

$$f(t) = f_0(t) + \Delta$$

Następnie za pomocą splotu funkcji $f(t) * g(t)$ otrzymać wygładzoną funkcję $f(t)$, $g(t)$ to funkcja wagowa.

Przyjmujemy parametry: $N_k = 2^k, k = 8, 10, 12$ – liczba węzłów, $T = 1.0, t_{max} = 3T$ – maksymalny okres czasu trwania sygnału, $dt = \frac{t_{max}}{N_k}$ – krok czasowy, $\sigma = \frac{T}{20}$, Δ - liczba pseudolosowa z przedziału $[-0.5, 0.5]$.

2. Opis metody

Każdą funkcję okresową, spełniającą warunki Dirichleta można rozwinąć w szereg trygonometryczny Fouriera

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{N} [a_k \cos(kx) + b_k \sin(kx)],$$

gdzie:

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt)\, dt,$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(kt)\, dt.$$

Szereg Fouriera w postaci zespolonej:

$$f(x) = \sum_{k=-n}^{n} c_k e^{Ikx},$$

gdzie:

$c_k = \frac{1}{2}(a_k - Ib_k).$

DFT (Dyskretna Transformata Fouriera) – przekształcenie funkcji $f(x)$ w ciąg $P(x)$ ortogonalnych jednomianów eksponencjalnych $E_k(x)$:

$$f(x) = P(x) = \sum_{k=0}^{N-1} < f, E_k > E_k,$$

gdzie:

$E_k(x) = e^{Ikx}, \qquad k = 0, \pm 1, \pm 2, \pm 3, \ldots$

FFT (Szybka Transformata Fouriera) – implementacja DFT, zmniejszająca liczbę niezbędnych operacji z $O(N^2)$ do $O(Nlog_2N)$.

Splot funkcji

Splot dwóch funkcji definiujemy jako:

$$(f * g) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau$$

Jeśli funkcję $f(t)$ potraktujemy jako sygnał, a funkcję $g(t)$ jako wagę, to splot obu funkcji możemy potraktować jako uśrednienie funkcji f pewną ustaloną funkcją wagową g. Wykorzystamy ten fakt do wygładzenia zaszumionego sygnału. Aby przeprowadzić efektywnie obliczenia, do obliczenia splotu wykorzystamy FFT:

$$FFT\{f(t) * g(t)\} = FFT\{f\} \cdot FFT\{g\} = f(k) \cdot g(k)$$

$$f * g = FFT^{-1}\{f(k) \cdot g(k)\}$$

Jako sygnał przyjmujemy:

$$f(t) = f_0(t) + \Delta$$

gdzie:

$$f_0(t) = \sin(1 \cdot \omega t) + \sin(2 \cdot \omega t) + \sin(3 \cdot \omega t)$$

jest sygnałem niezaburzonym,

$\omega = \frac{2\pi}{T}$ - pulsacja,

T – okres,

Δ - liczba pseudolosowa z zakresu $\left[-\frac{1}{2}, \frac{1}{2}\right]$.

Jako funkcję wagową przyjmujemy funkcję gaussowską:

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

Ponieważ będziemy operować dla chwil czasowych $t \in [0, t_{max}]$, więc funkcja $g(t)$ będzie tylko „połówką” pełnej funkcji gaussowskiej (ponieważ środek wypada w $t = 0$). Dlatego musimy dodać drugą „połówkę”.

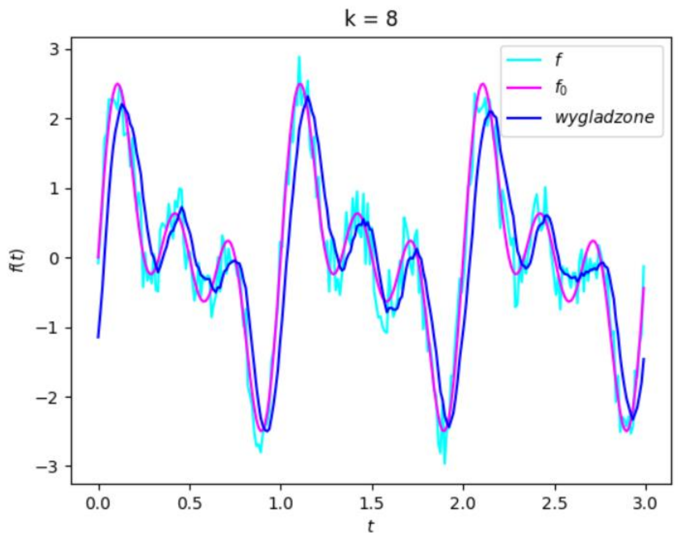
Zamiast $g(k) = FFT\{g(t)\}$ do liczenia splotu musimy użyć sumy dwóch transformat:

$$g(k) = g_1(k) + g_2(k) = FFT\{g(t)\} + FFT^{-1}\{g(t)\}.$$

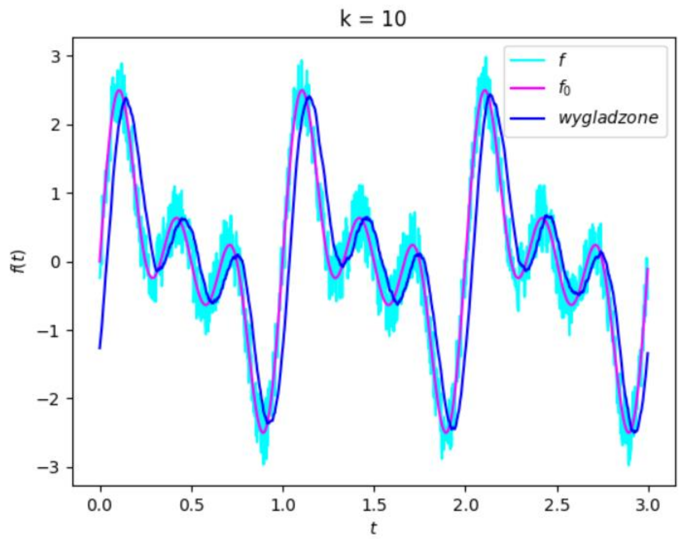
3. Wykresy i wyniki

W Pythonie za pomocą biblioteki matplotlib narysowaliśmy wykresy porównania oryginalnego sygnału niezaburzonego, sygnału zaburzonego oraz odszumionego dla funkcji:

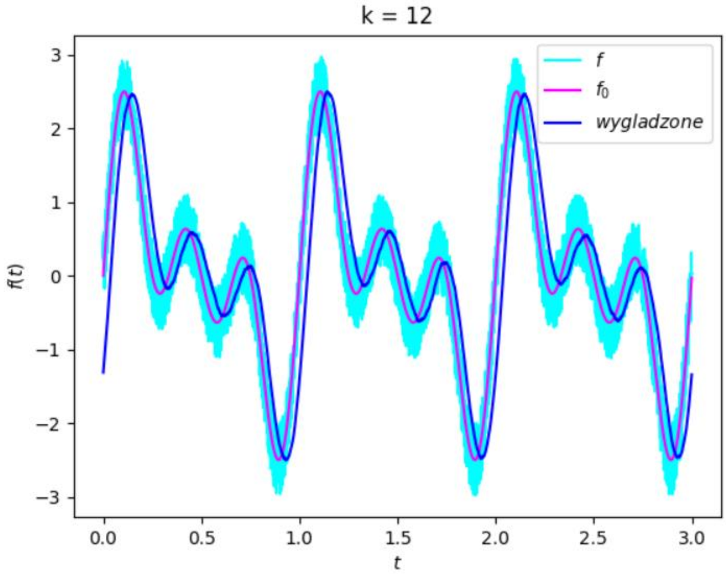
$$f_0(t) = \sin(1 \cdot \omega t) + \sin(2 \cdot \omega t) + \sin(3 \cdot \omega t)$$



Rys.1 *k = 8, $N_k = 2^8$*



Rys.2 *k = 10, $N_k = 2^{10}$*



Rys.2 $k = 12$, $N_k = 2^{12}$

Na Rys.1 dla *k = 8* kształt funkcji splotu jest zbliżony do prawidłowego, jednak nie jest on gładki.

Dla *k = 10* wykres jest bardziej wygładzony, a dla *k = 12* wykres jest prawie idealnie gładki.

4. Wnioski

Odszumianie sygnału przy użyciu splotu funkcji, do którego wyznaczenia użyto FTT, jest dobrym narzędziem, dającym zazwyczaj wystarczająco satysfakcjonujące rezultaty. Na wygenerowanych wykresach widzimy, że aby zmaksymalizować gładkość funkcji, liczba węzłów powinna być jak największa.

Bibliografia

http://home.agh.edu.pl/~chwiej/mn/fft_1819.pdf