# Sprawozdanie

## Metody numeryczne

### Laboratorium 11.

*Odszumianie sygnału przy użyciu szybkiej transformacji Fouriera (FFT – Fast Fourier Transform) - splot funkcji*

27.05.2020 r.

Aleksandra Rolka

# 1. Wstęp teoretyczny

## Szybka transformacja Fouriera

Szybka transformacja Fouriera (ang. Fast Fourier Transform, FFT) – algorytm wyznaczania dyskretnej transformaty Fouriera oraz transformaty do niej odwrotnej. Najpopularniejszą wersją algorytmu FFT jest FFT o podstawie 2. Jest on bardzo efektywny pod względem czasu realizacji, jednak wektor próbek wejściowych (spróbkowany sygnał) musi mieć długość N = $2^k$, gdzie k to pewna liczba naturalna.
Dzięki szybkiej transformacie Fouriera praktycznie możliwe stało się cyfrowe przetwarzanie sygnałów (DSP), a także zastosowanie dyskretnych transformat kosinusowych (DCT) do kompresji danych audio-wideo (JPEG, MP3, XviD itd.).

## Algorytm radix-2

Najprostszy algorytm FFT to radix-2 (Cooley-Tukey) opracowany w latach 60 XX. wieku w celu szybkiej analizy danych sejsmologicznych.

Założenia algorytmu:

- $x_j = \frac{2\pi}{N} j$ , (1)
- $j = 0, 1, 2, \dots, N-1$ , (2)
- $N = 2^r, r \in N$ . (3)

Współczynniki transformaty Fouriera (DFT)$c_k$ wyznaczane są następująco:

$$c_k = \langle E_k, f \rangle = \sum_{j=0}^{N-1} E^*{}_k(x_j) f(x_j) = \sum_{j=0}^{N-1} f(x_j) \exp(-I\, x_j k) = \sum_{j=0}^{N-1} f_j \exp(-I \frac{2\pi}{N} jk) \quad (4)$$

Grupowane osobno są składniki parzyste ( $j = 2m$) i nieparzyste ( $j = 2m + 1$):

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp(-I \frac{2\pi}{N}(2m)k) + \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp(-I \frac{2\pi}{N}(2m+1)k) , \quad (5)$$

równoważne:

$$c_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp(-I \frac{2\pi}{N/2} m\, k) + \exp\left(-I \frac{2\pi}{N} k\right) \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp(-I \frac{2\pi}{N/2} m\, k). \quad (6)$$

Oznaczając:

$$p_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \exp\left(-I \frac{2\pi}{N/2} m\, k\right), \tag{7}$$

$$q_k = \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \exp\left(-I \frac{2\pi}{N/2} m\, k\right), \tag{8}$$

$$\varphi_k = \exp\left(-I \frac{2\pi}{N} k\right), \tag{9}$$

wzór (6) można zapisać w postaci:

$$c_k = p_k + \varphi_k q_k. \tag{10}$$

Korzystając z okresowości wyrazów $p_k$ oraz $q_k$:

$$p_{k+N/2} = p_k, \tag{11}$$

$$q_{k+N/2} = q_k, \tag{12}$$

$$\varphi_{k+N/2} = -\varphi_k, \tag{13}$$

(nie ma potrzeby wyznaczać wszystkich współczynników – tylko połowę)

gdzie:

- współczynniki $p_k$ oraz $q_k$ można wyliczyć dzięki DFT nakładem $O\left(\frac{N}{2}\right)^2 = O(\frac{N^2}{4})$,
- dodatkowo zaoszczędzić można czas wyznaczając tylko współczynniki dla $k < \frac{N}{2}$, ponieważ:

$$c_k = \begin{cases} p_k + \varphi_k q_k, & k < \frac{N}{2} \\ p_{k-\frac{N}{2}} - \varphi_k q_{k-\frac{N}{2}}, & k \geq \frac{N}{2} \end{cases}. \tag{14}$$

Następnym krokiem FFT jest podział sum w $p_k$ oraz $q_k$ na sumy zawierające tylko elementy parzyste i nieparzyste. Po podziale liczba elementów w każdej z dwóch powstałych sum jest dwukrotnie mniejsza niż w elemencie macierzystym.
Proces rekurencyjnego podziału zostaje zakończony, gdy liczba elementów jest równa 1.

# 2. Zadanie do wykonania

## 2.1 Opis problemu

Zdefiniowano splot funkcji jako:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau, \tag{15}$$

Funkcja $f(t)$ jest sygnałem, a funkcja $g(t)$ wagą, więc splot (15) można potraktować jako uśrednienie funkcji $f$ pewną ustaloną funkcją wagową g. Należało wykorzystać ten fakt do

wygładzenia zaszumionego sygnału.
Do przeprowadzenia efektywnych obliczeń do obliczenia splotu wykorzystano FFT.

Jako sygnał przyjęto:

$$f(t) = f_0(t) + \Delta, \tag{16}$$

gdzie $f_0$ jest sygnałem niezaburzonym (bez szumu) zdefiniowanym następująco:

$$f_0(t) = \sin(1 \cdot \omega t) + \sin(2 \cdot \omega t) + \sin(3 \cdot \omega t), \tag{17}$$

$\omega = 2\pi/T$ – pulsacja,
$T$ – okres ,
$\Delta$ – liczba pseudolosowa z przedziału $\left[-\frac{1}{2}, \frac{1}{2}\right]$.

Jako funkcję wagową przyjęto funkcję gaussowską:

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right) \tag{18}$$

W zadaniu przyjęto następujące parametry:

- $N = 2^k$ – całkowita liczba węzłów, gdzie $k = 8, 10, 12$,
- $T = 1.0$ ,
- $t_{max} = 3T$ – maksymalny okres rejestracji sygnału,
- $dt = \frac{t_{max}}{N}$ – krok czasowy,
- $\sigma = \frac{T}{20}$.

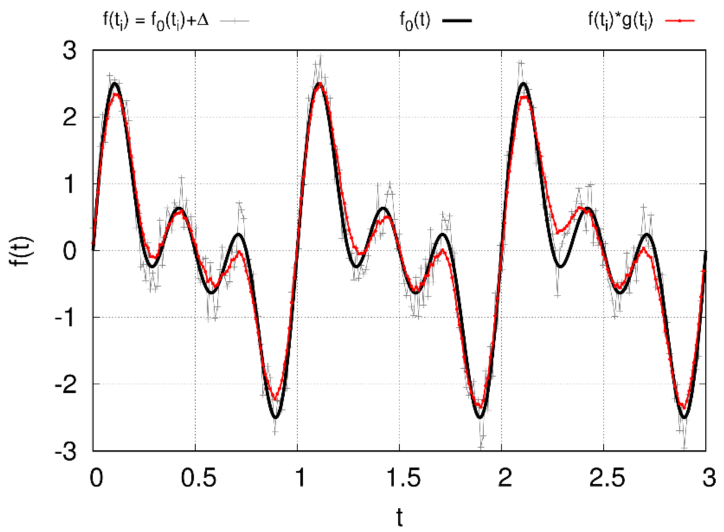
## 2.2 Wyniki

Napisany program rozwiązujący powyższy problem korzysta z dodatkowej biblioteki numerycznej *GSL*. Do obliczenia FFT wykorzystano procedurę *gsl_fft_complex_radix2_forward*, a do wyznaczenia transformaty odwrotnej procedurę *gsl_fft_complex_radix2_backward*.

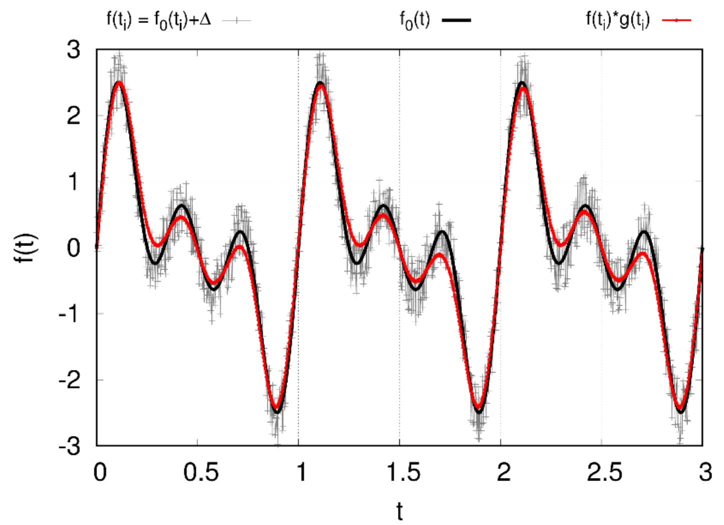
Otrzymane wyniki obliczeń zapisane zostały w plikach **.dat:*

- *k(nr).dat* – trzy pliki z odpowiedniki numerami odpowiadającymi wartości *k*,
  w każdym z nich znajdują się dwie serie danych: sygnał oryginalny, zaszumiony
  oraz odszumiony sygnał wraz z daną chwilą czasową

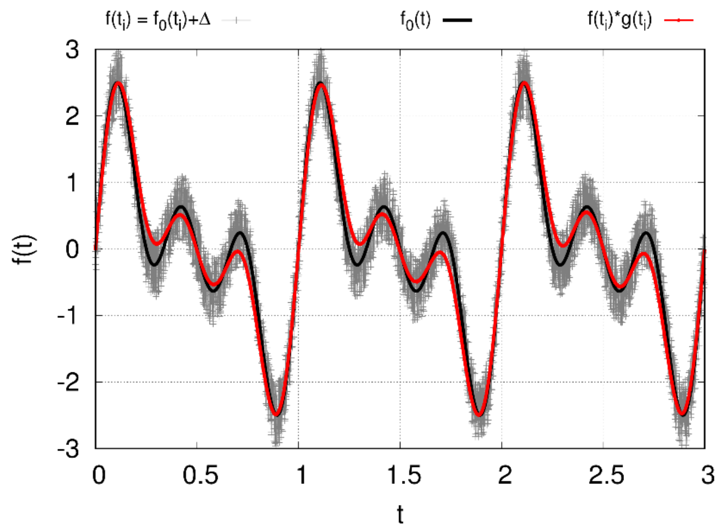
Następnie na podstawie tych danych za pomocą skryptu Gnuplot’a wygenerowano poniższe wykresy:




$(a)\ k = 8 \implies N_k = 2^8$ próbek wejściowych




$(b)\ k = 10 \implies N_k = 2^{10}$ próbek wejściowych



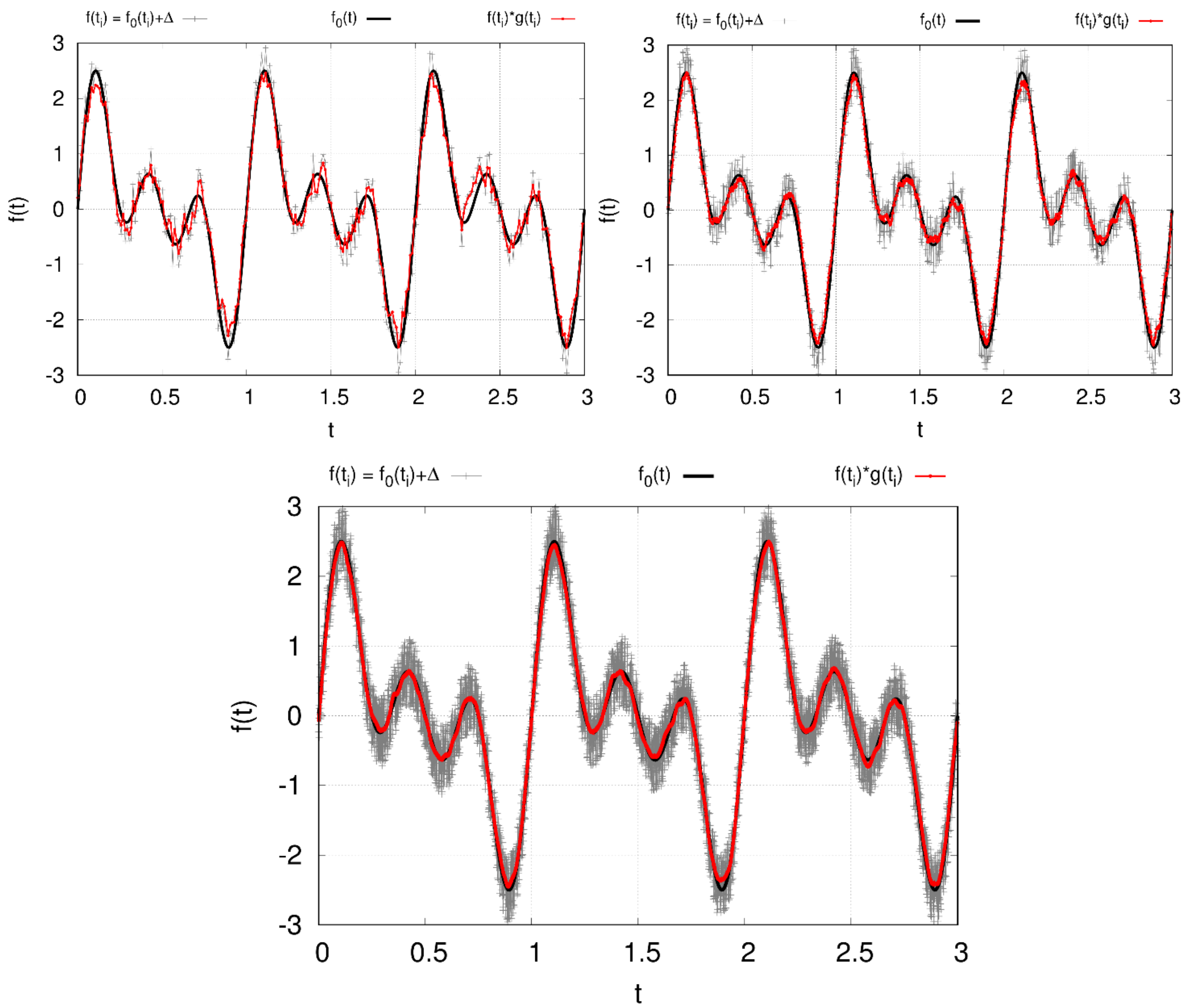

$(c)\ k = 12 \implies N_k = 2^{12}$ próbek wejściowych

Rysunek 1. *Wynik odszumiania sygnału przy użyciu FFT; f0(t) – oryginalny sygnał niezaburzony, f(t) = f0(t) + Δ – sygnał zaburzony, f(t) ∗ g(t) – sygnał wygładzony (odszumiony), tj. splot funkcji.*

# 3. Wnioski

Szybka transformacja Fouriera nie pozwoliła na uzyskanie idealnego odwzorowania funkcji oryginalnej.
Na wykresie dla *k=8*,czyli *N=2⁸* widoczny jest ogólny kształt oczekiwanej funkcji jednak funkcja splotu nie jest gładka. Ulega ona wygładzeniu przy zwiększeniu *N* do równego $2^{10}$, dopasowanie również ulega poprawie. Najlepszy rezultat widoczny jest dla *k=12*, czyli $N=2^{12}$. Funkcja splotu jest najgładsza i najbardziej przypomina sygnał niezaburzony, jednak dopasowanie wciąż nie jest dokładne. Jedynie w ekstremach globalnych zaobserwowano niemalże idealne pokrycie się funkcji. Zaobserwowano również, że największe różnice w dopasowaniu były pomiędzy pierwszym (a), a drugim (b) przypadkiem, pomiędzy drugim (b), a trzecim (b) nie były już tak widoczne.

Dla porównania uruchomiono również program dla wartości $\sigma = T/100$ :




Rysunek 2. *Wynik odszumiania sygnału przy użyciu FFT przy* $\sigma = \frac{T}{100}$*; f0(t) – oryginalny sygnał niezaburzony, f(t) = f0(t) + Δ – sygnał zaburzony, f(t) $*$ g(t) – sygnał wygładzony (odszumiony), tj. splot funkcji. Kolejno dla k=8, 10, 12.*

Widać, że przy zmniejszeniu wartości odchylenia standardowego funkcji wagowej uzyskuje się dokładniejsze ogólne dopasowanie funkcji, jednak jest to kosztem dużo gorszego wygładzenia.

Szybka transformata Fouriera jest bardzo dobrym narzędziem do odszumienia funkcji zaszumionego sygnału, jednak nie daje ona idealnego odwzorowania funkcji sygnału niezaburzonego, a raczej zadowalające przybliżenie.

Źródła:


- Dr hab inż. Tomasz Chwiej – Notatki do wykładu *„Szybka transformacja Fouriera”*. Dostępne w Internecie: http://galaxy.agh.edu.pl/~chwiej/mn/fft_1819.pdf
- Szybka transformacja Fouriera – Wikipedia.org. Dostępne w Internecie: https://pl.wikipedia.org/wiki/Szybka_transformacja_Fouriera