Sprawozdanie - Laboratorium 11
Odszumianie sygnału przy użyciu FFT - splot funkcji

Marta Dychała

24 maja 2021

1 Wstęp teoretyczny

Istnieje wiele metod pozwalających na dokonanie interpolacji bądź aproksymacji funkcji w zależności od jej typu. Jeżeli rozważana funkcja jest okresowa, to w tym celu przydatne jest wykorzystanie jej rozwinięcia szeregu Fouriera.

Jeden z najprostszych przypadków szeregu Fouriera występuje dla funkcji, której okres wynosi $T = 2\pi$. Wówczas funkcję tą można rozwinąć w szereg Fouriera postaci:

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

gdzie:

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

Szereg Fouriera opisany wzorem (1) można przedstawić w postaci zespolonej:

$$f(x) = \sum_{k=-n}^{n} c_k e^{Ikx}, \tag{2}$$

gdzie $c_k = \frac{1}{2}(a_k - Ib_k)$. Zamieniając czynnik e^{Ikx} na funkcję $E_k(x)$:

$$E_k(x) = e^{Ikx}, \qquad k = 0, \pm1, \pm2, ...$$

możliwe staje się utworzenie „wielomianu” z szeregu (2), w którym „jednomianem” jest funkcja $E_k(x)$:

$$P(x) = \sum_{k=0}^{N-1} c_k e^{Ikx} = \sum_{k=0}^{N-1} c_k E_k(x)$$

Jeżeli wartości funkcji f są określone na siatce złożonej z równoodległych węzłów, to wielomian $P(x)$ może posłużyć do wyznaczenia **dyskretnej transformacji Fouriera** (DFT, ang. *Discrete Fourier Transform*):

$$f(x) = P(x) = \sum_{k=0}^{N-1} \langle f, E_k \rangle E_k \tag{3}$$

Za pomocą DFT można dokonywać aproksymacji i interpolacji funkcji okresowych, jednakże sama operacja ma dużą złożoność obliczeniową rzędu $O(n^2)$. Dlatego też powstał szybszy sposób na wyznaczanie szeregu Fouriera danej funkcji, zwany **szybką transformacją Fouriera** (FFT, ang. *Fast Fourier Transform*). Metoda ta ma znacznie niższą złożoność - rzędu zaledwie $O(N \log_2 N)$.
Podstawowym algorytmem na jakim bazuje FFT jest algorytm radix-2 zwany także algorytmem Cooley'a-Tukey'a. Głównym jego założeniem jest liczba węzłów, która jest potęgą liczby 2. Działa na zasadzie „dziel i zwyciężaj” - główny problem dzieli się rekurencyjnie na dwa podproblemy, które są rozwiązywane w celu znalezienia całkowitego rozwiązania. Szybka transformacja Fouriera posiada szerokie zastosowania w

dziedzinie techniki - używa się jej przy kompresji danych ale także w przetwarzaniu sygnałów cyfrowych. W tym celu przy pomocy FFT można przeprowadzić całkowanie numeryczne funkcji za pomocą splotu, a otrzymana funkcja w tym procederze będzie przybliżeniem niezaburzonego sygnału.

Splot funkcji f i g zdefiniowany jest jako:

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

Jest on funkcją otrzymanego uśrednionego sygnału $f(t)$ za pomocą wagi $g(t)$. Można go obliczyć wykorzystując FFT:

$$FFT\{f(t) * g(t)\} = FFT\{f\} \cdot FFT\{g\} = f(k) \cdot g(k)$$
$$f * g = FFT^{-1}\{f(k) \cdot g(k)\} \tag{4}$$

2 Zadanie do wykonania

2.1 Opis problemu

Celem zajęć laboratoryjnych było wyznaczenie przybliżenia funkcji odszumionego sygnału wykorzystując w tym celu szybką transformację Fouriera. Sygnał był określony wzorem:

$$f(t) = f_0(t) + \Delta, \tag{5}$$

gdzie Δ to liczba pseudolosowa z przedziału $[-0.5, 0.5]$, natomiast $f_0(t)$ to szukany odszumiony sygnał, którego wzór był postaci:

$$f_0(t) = \sin(\omega t) + \cos(2\omega t) + \sin(3\omega t), \tag{6}$$

gdzie $\omega = \frac{2\pi}{T}$ - pulsacja a T to okres sygnału. Jako funkcję wagową g przyjęto funkcję Gaussa:

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right) \tag{7}$$

Odszumianie sygnału zrealizowano dla trzech przypadków: $k = \{8, 10, 12\}$. W każdym z nich przyjęto poniższe parametry:

- $N_k = 2^k$ - liczba węzłów
- $T = 1$ - okres trwania sygnału
- $t_{max} = 3T$ - maksymalny okres trwania sygnału
- $dt = t_{max}/N_k$ - krok czasowy
- $\sigma = T/20$ - odchylenie standardowe funkcji wagowej

Do zrealizowania zadania wykorzystano bibliotekę Numerical Recipes w języku C. Każda funkcja w programie była reprezentowana przez wektor o rozmiarze $2N_k$. Jako nieparzyste indeksy przyjmowały one części rzeczywiste wartości funkcji, natomiast jako parzyste - urojone.

Ponieważ zastosowanie algorytmu FFT na funkcji g skutkowałoby w znalezieniu jedynie „połówki" funkcji gausssowskiej (rozważany przedział to $t \in [0, t_{max}]$), funkcja g została początkowo podzielona na dwie funkcje - g_1 oraz g_2, będącymi jej „połówkami". Z tego powodu funkcja $g(t)$ była sumą transformaty Fouriera funkcji g_1 oraz transformaty odwrotnej Fouriera funkcji g_2:

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

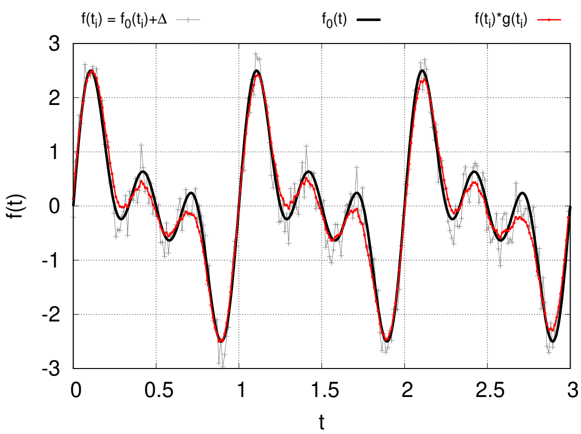
Na początku wypełniono tablice odpowiadające funkcjom $f_0(t)$, $f(t)$, $g_1(t)$ oraz g_2 w zależności od kroku czasowego $t \in [0, t_{max}]$, gdzie $t = dt \cdot (i-1)$ dla elementów o nieparzystych indeksach i oraz 0 dla parzystych indeksów. Następnie w celu obliczenia transformat f i g_1 jak i transformaty odwrotnej g_2 wykorzystano procedurę `four1()` pochodzącą z biblioteki Numerical Recipes. Po wyznaczeniu postaci funkcji g, wyznaczono splot funkcji f i g zgodnie ze wzorem (4), nadpisując przy tym tablicę f otrzymanym splotem.

Obliczone wartości funkcji f zaszumionej (wzór (5)) oraz otrzymany splot (wzór (4)) po normalizacji, czyli podzieleniu wszystkich wartości funkcji przez $\frac{f_{max}}{2.5}$, gdzie f_{max} to wartość największa splotu, zapisano do plików, które posłużyły do wygenerowania wykresów w programie Gnuplot.

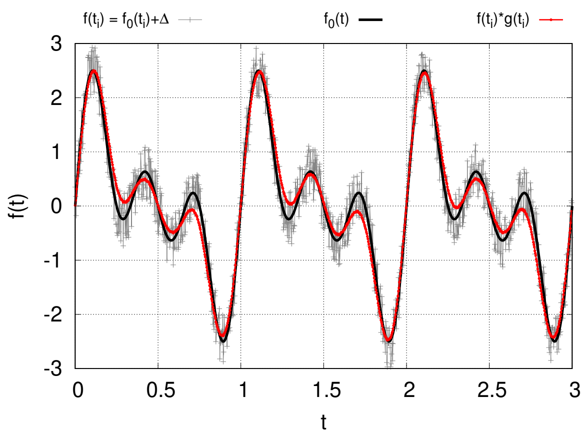
2.2 Wyniki

Na podstawie danych z utworzonych plików sporządzono wykresy w celu porównania otrzymanego numerycznie splotu funkcji f i g z teoretycznym odszumieniem, określonego na podstawie wzoru (6). Na każdym z rysunków znajdują się trzy rodzaje wykresów oznaczone różnymi kolorami:

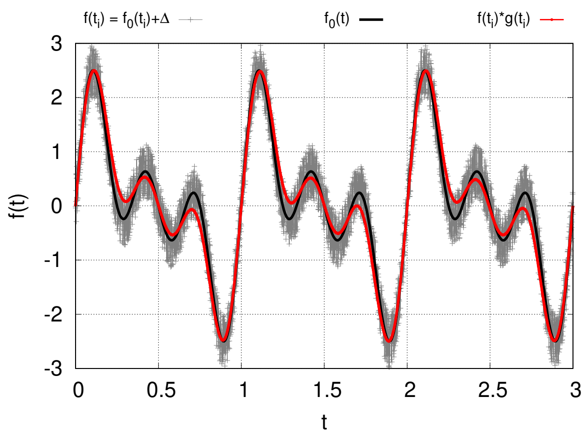
- **czerwonym kolorem** została oznaczona funkcja splotu f wyliczona ze wzoru (4),
- **szarym kolorem** oznaczono zaszumiony sygnał f obliczony ze wzoru (5),
- **czarnym kolorem** oznaczono funkcję f_0 będącą odszumieniem sygnału wynikającym z teorii, opisaną wzorem (6).



(a) $k = 8 \Longrightarrow N_k = 2^8 = 256$



(b) $k = 10 \Longrightarrow N_k = 2^{10} = 1024$

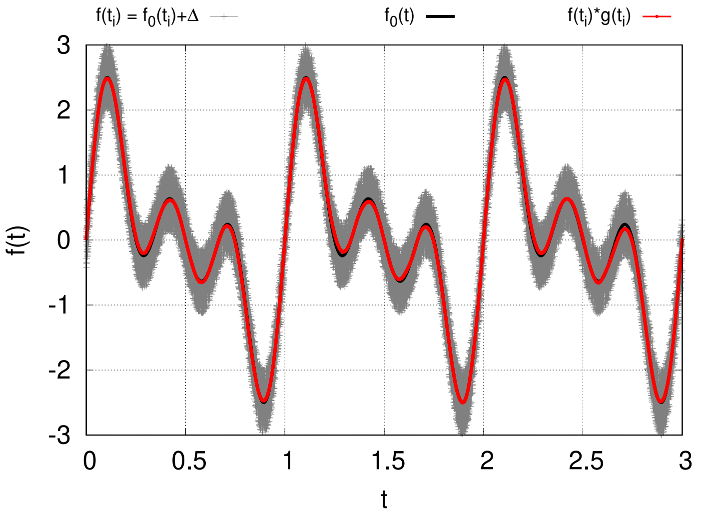


(c) $k = 12 \Longrightarrow N_k = 2^{12} = 4096$

Rysunek 1: Wyniki odszumiania sygnału przy użyciu FFT w zależności od parametru k.

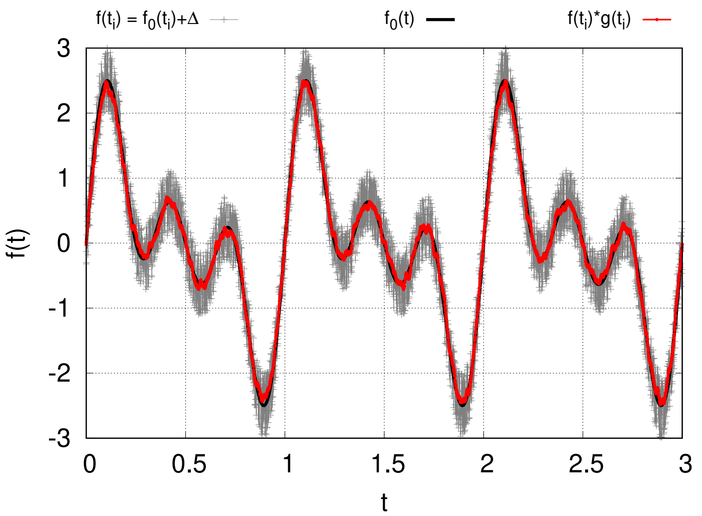
Na wykresach powyżej można zauważyć, że wraz ze wzrostem parametru k otrzymana funkcja splotu jest coraz bardziej gładka. Dla $k = 8$ w niektórych miejscach funkcji splotu widoczne są nadmiarowe ekstrema, zanikają one jednak przy większych wartościach k. Można zaobserwować, że przy zastosowaniu większej liczby węzłów dopasowywana funkcja staje się gładsza. Okazuje się, by uzyskać funkcję o satysfakcjonującej gładkości, wystarczy przyjąć, że $k = 12$, bowiem im większa jest wartość parametru k, tym trudniej jest zauważyć różnicę w określeniu kształtu funkcji. Różnica między wykresem (a) i (b) jest lepiej widoczna niż pomiędzy wykresami (b) i (c).

Pomimo, że dla $k = 12$ kształt funkcji splotu przypomina kształt funkcji f_0, to jednak otrzymana funkcja f nie jest dokładnym przybliżeniem funkcji f_0, co szczególnie jest widoczne w okolicach ekstremów lokalnych. Problem ten jest związany głównie z odchyleniem standardowym σ funkcji wagowej - w trakcie wykonywania ćwiczenia ustalono go na zbyt dużą wartość. Zmniejszenie wartości σ np. do $T/55$ znacznie poprawia przebieg odszumionej funkcji:



Rysunek 2: Odszumienie sygnału przy użyciu FFT dla $k = 12$ oraz $\sigma = T/55$

Ważne jest także, aby wartość parametru σ nie była również zbyt mała, ponieważ wówczas otrzymana funkcja splotu staje się zbyt podatna na pojedyncze odchylenia wartości od średniej. Oznacza to, że aby uzyskać gładką funkcję sygnału, potrzeba użyć większej liczby węzłów, co przekłada się na większe zużycie pamięci i czasochłonność algorytmu.



Rysunek 3: Odszumienie sygnału przy użyciu FFT dla $k = 12$ oraz $\sigma = T/200$

Własność ta powoduje, że przez zbyt małe wartości σ trudniej jest odzyskać pierwotną funkcję sygnału.

3 Wnioski

Szybka transformacja Fouriera jest przydatnym narzędziem do znajdowania przybliżeń funkcji okresowych. Okazuje się, że już dla niewielkich wartości k można uzyskać funkcję z zadowalającym wygładzeniem. Ważne jednak okazuje się także dobranie odpowiedniej wartości parametru σ - w treści zadania był on za duży, co sprawiło, że otrzymany splot funkcji nie odzwierciedlał zbyt dokładnie przebiegu funkcji f_0. Optymalną wartością parametru σ w stosunku do pozostałych danych w zadaniu okazuje się być $\sigma \in [T/65, T/45]$.