Wydział: Fizyki i Informatyki Stosowanej
Kierunek: Informatyka Stosowana
Rok: 2020/21
Semsetr: letni
Typ: stacjonarne
Nr albumu: 401984
Data: 24.05.2021

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

Spis treści

opracował:
Tomasz Szkaradek

1 Wstęp teoretyczny

1.1 Szereg i Transformacja Fouriera

Szereg Fouriera - szereg pozwalający rozłożyć funkcję okresową, spełniającą warunki Dirichleta, na sumę funkcji trygonometrycznych. Szeregi Fouriera zostały wprowadzone w 1807 roku przez Josepha Fouriera w celu rozwiązania równania ciepła dla metalowej płyty. Dziś mają one wielkie znaczenie między innymi w fizyce, teorii drgań oraz przetwarzaniu sygnałów obrazu (kompresja jpeg) i dźwięku (kompresja mp3).
Natomiast transformacja Fouriera przetwarza funkcję z danej przestrzeni w ten sposób, że wyeksponowane są jej własności okresowe i częstotliwościowe. Transformacja ta jest podstawowym narzędziem analizy harmonicznej i teorii analizy i przetwarzania sygnału. Jako, że jest to narzędzie gwarantujące dokładność i bezstratność przekształcenia, można nie tylko wyznaczać samą transformatę funkcji, ale również transformatę odwrotną, z równie dokładnym wynikiem. Jeśli funkcja $f(x)$ jest okresowa możemy ją rozwinąć w Szereg Fouriera postaci:

$$\begin{gathered} f(x) = \tfrac{a_0}{2} + \sum_{k=1}^{\infty}(a_k cos(kx) + b_k sin(kx)) \\ a_k = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(t)cos(kt)dt \\ b_k = \tfrac{1}{\pi}\int_{-\pi}^{\pi} f(t)sin(kt)dt \end{gathered} \tag{1}$$

Funkcję możemy też zapisać w postaci Zespolonego Szeregu Fouriera:

$$\begin{gathered} f(x) \sim \sum_{k=-\infty}^{\infty} \hat{f}(x)e^{Ikx} \\ \hat{f}(x) = \tfrac{1}{2\pi}\int_{-\pi}^{\pi} f(t)e^{-Ikt}dt \end{gathered} \tag{2}$$

Niech $x_0, \cdots, x_{N-1}$ będą liczbami zespolonymi, wtedy dyskretna transformata Fouriera (DFT) jest określona wzorem

$$X_k = \sum_{n=0}^{N-1} x_n exp(\frac{2\pi i}{N} nk), \quad k = 0, \cdots, N-1 \tag{3}$$

gdzie
i – jednostka urojona,
k – numer harmonicznej,
n – numer próbki sygnału,
x_n – wartość próbki sygnału,
N – liczba próbek.

1.2 Szybka transformata Fouriera (FFT)

Szybka transformacja Fouriera (FFT) to algorytm wyznaczania dyskretnej transformaty Fouriera oraz transformaty do niej odwrotnej.
Algorytm ten można wykorzystać m.in. do cyfrowego przetwarzania sygnałów (DSP) oraz kompresji danych audio-wideo. Liczenie transformaty Fouriera z definicji jest wyjątkowo pracochłonne - złożoność obliczeniowa wynosi aż $O(n^2)$, gdzie algorytmy szybkiej transformacji Fouriera, opierające się w znakomitej większości na dzieleniu transformaty na mniejsze (zgodnie z regułą "dziel i zwyciężaj") potrafią sprowadzić złożoność do poziomu $O(nlog(n))$, co jest niemałą różnicą przy przetwarzaniu obszernych danych. Głównie dzięki wydajności, jaką się charakteryzuje FFT, została uznana za jeden z najważniejszych algorytmów numerycznych opracowanych w XX wieku. Jest to zagadnienie wysoce istotne, do tego stopnia, że nadal trwają próby osiągnięcia niższej złożoności obliczeniowej, by jeszcze bardziej usprawnić ten algorytm.
Na tych laboratoriach zajęliśmy się jednak, odszumianiem sygnału przy użyciu FFT, co jest kolejnym z zastosowań tego narzędzia.

1.3 Rozkład normalny, rozkład Gaussa

Rozkład normalny, rozkład Gaussa to jeden z najważniejszych rozkładów prawdopodobieństwa, odgrywający ważną rolę w statystyce. Wykres funkcji prawdopodobieństwa tego rozkładu jest krzywą w kształcie dzwonu (tak zwaną krzywą dzwonową).
Przyczyną jego znaczenia jest częstość występowania w naturze. Jeśli jakaś wielkość jest sumą lub średnią bardzo wielu drobnych losowych czynników, to niezależnie od rozkładu każdego z tych czynników jej rozkład będzie zbliżony do normalnego (centralne twierdzenie graniczne) - dlatego można go bardzo często zaobserwować w danych. Ponadto rozkład normalny ma interesujące właściwości matematyczne, dzięki którym oparte na nim metody statystyczne są proste obliczeniowo. Funkcja gęstości prawdopodobieństwa rozkładu normalnego jest przykładem funkcji Gaussa. Dana jest ona wzorem:

$$f_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} exp^{(\frac{-(x-\sigma)^2}{2\sigma^2})} \tag{4}$$

gdzie
μ to średnia
σ to odchylenie standardowe
σ^2 to wariancja
Fakt, iż zmienna losowa X ma rozkład normalny z wartością oczekiwaną μ i wariancją σ^2 zapisuje się często $X \sim N(\mu, \sigma)$. Jeśli μ=0 i $\sigma = 1$ to rozkład ten nazywa się standardowym rozkładem normalnym, jego funkcja gęstości opisana jest wzorem:

$$\phi_{0,1}(x) = \phi(x) = \frac{1}{\sqrt{2\pi}} exp^{(\frac{-x^2}{2})} \tag{5}$$

2 Zadanie do wykonania

Naszym zadaniem w trakcie laboratoriów było zastosowanie algorytmu FFT do odszumiania sygnału. Sygnał okresowy, niezaszumiony, miał postać:

$$f_0(t) = sin(1 \cdot \omega t) + sin(2 \cdot \omega t) + sin(3 \cdot \omega t) \tag{6}$$

gdzie i $= 0, 1, \cdots, N_k$ natomiast $N_k = 2^k$ ilość wygenerowanych próbek sygnału.
k = 8, 10, 12
$\omega = \dfrac{2\pi}{T}$ - pulsacja,
T - okres
Sygnał zaszumiony miał być wygenerowany w następujący sposób.

$$f(t) = f(t) + \Delta \tag{7}$$

Δ jest liczbą pseudolosową z zakresu [-1/2, 1/2] jest wyznaczana dla każdego indeksu i z osobna.
Jako funkcję wagową przyjmiemy funkcje gaussowską:

$$g(t) = \frac{1}{\sigma\sqrt{2\pi}} exp^{(\dfrac{-t^2}{2\sigma^2})} \tag{8}$$

Splot funkcji zdefiniowany jest jako:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)(t - \tau) \tag{9}$$

Przyjmując f(t) jako sygnał a funkcję g(t) jako wagę, to splot obu funkcji można potraktować jako uśrednienie funkcji f pewną ustaloną funkcją wagową g. Fakt ten zostanie wykorzystany do wygładzenia zaszumionego sygnału. W tym celu posłużymy się szybką transformacją Fouriera:

$$FFT\{f(t) * g(t)\} = FFT\{f\} \cdot FFT\{g\} = f(k) \cdot g(k) \tag{10}$$

$$f * g = FFT^{-1}\{f(k) \cdot g(k)\} \tag{11}$$

Ponieważ będziemy operować dla chwil czasowych t $\in$ [0, 3T] więc funkcja g(t) będzie tylko "połówką" pełnej funkcji gaussowskiej (ponieważ jej środek wypada w t = 0. Dlatego w obliczeniach musimy dodać drugą "połówkę". Licząc $g_1(k)$ stosujemy wzór:
$g_1(k) = FFTg(t > 0)$
Natomiast licząc $g_2(k)$ musimy zmienić znak przy t g(t) = g(-t) ze względu na symetrię:
$g_2(k) = FFTg(t < 0)$

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,
$\sigma = \frac{T}{20}$
Tworzymy pętlę zewnętrzną po k = 8, 10, 12, wyznaczamy w niej N_k, i tworzymy tablice z wartościami dla f_0 oraz f (o długości N_k):
Pierwszym naszym zadaniem było odpowiednie wypełnienie tablic wartościami, policzenie kolejnych transformat i transformaty odwrotnej:

$$f_k = FFT\{f\}, \quad g_1(k) = FFT\{g_1\}, \quad g_2(k) = FFT\{g_2\} \tag{12}$$

a następnie wyznaczenie transformaty splotu

$$(f_k(g_1(k) + g_2(k)) \tag{13}$$

Po tym kroku nadpisaliśmy tablicę f transformatą splotu, a następnie obliczyliśmy następującą transformatę odwrotną:

$$FFT^{-1}\{f(k)\} \tag{14}$$

uzyskując w ten sposób w tablicy f wygładzoną funkcję f(t)
Później dla tablicy f musimy znaleźć element o maksymalnym module f_{max} a następnie z jego pomocą znormalizować wartości w tablicy f:

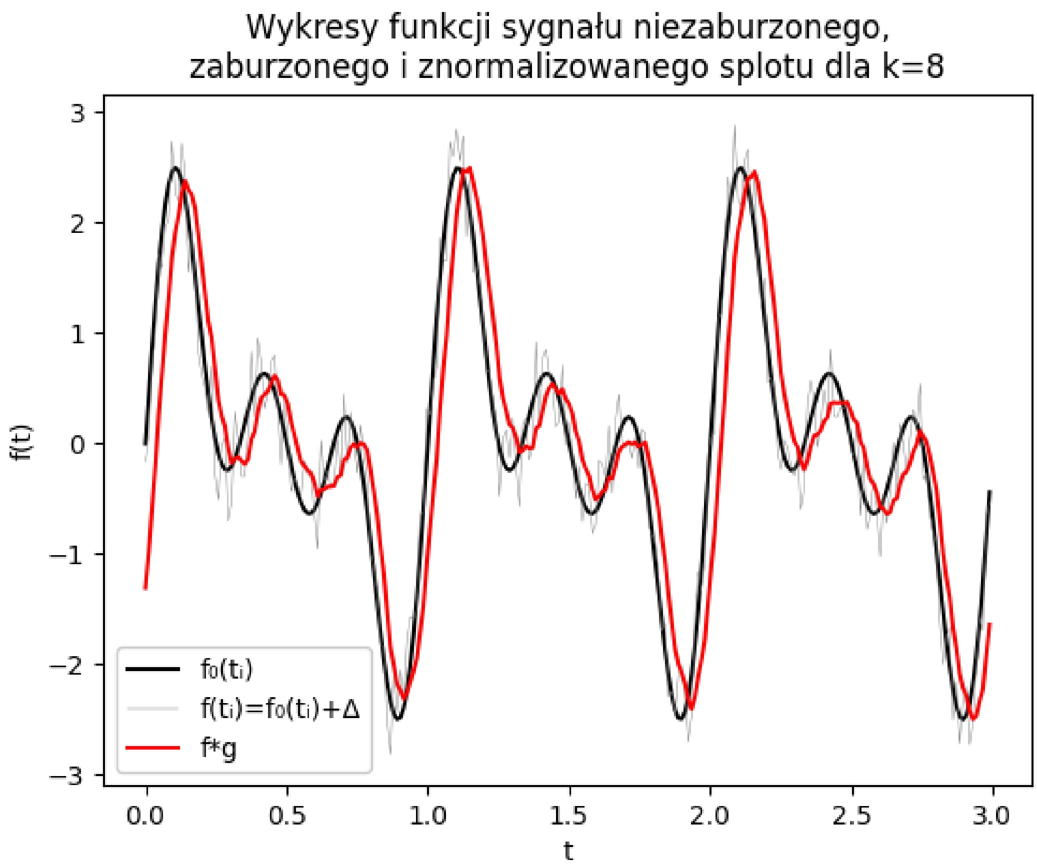
$$f \cdot \frac{2.5}{f_{max}} \tag{15}$$

Na sam koniec nanosimy nasze wartości na wykres to jest funkcja f_0 niezaburzona wartości funkcji f zaszumioną oraz wartości dla splotu

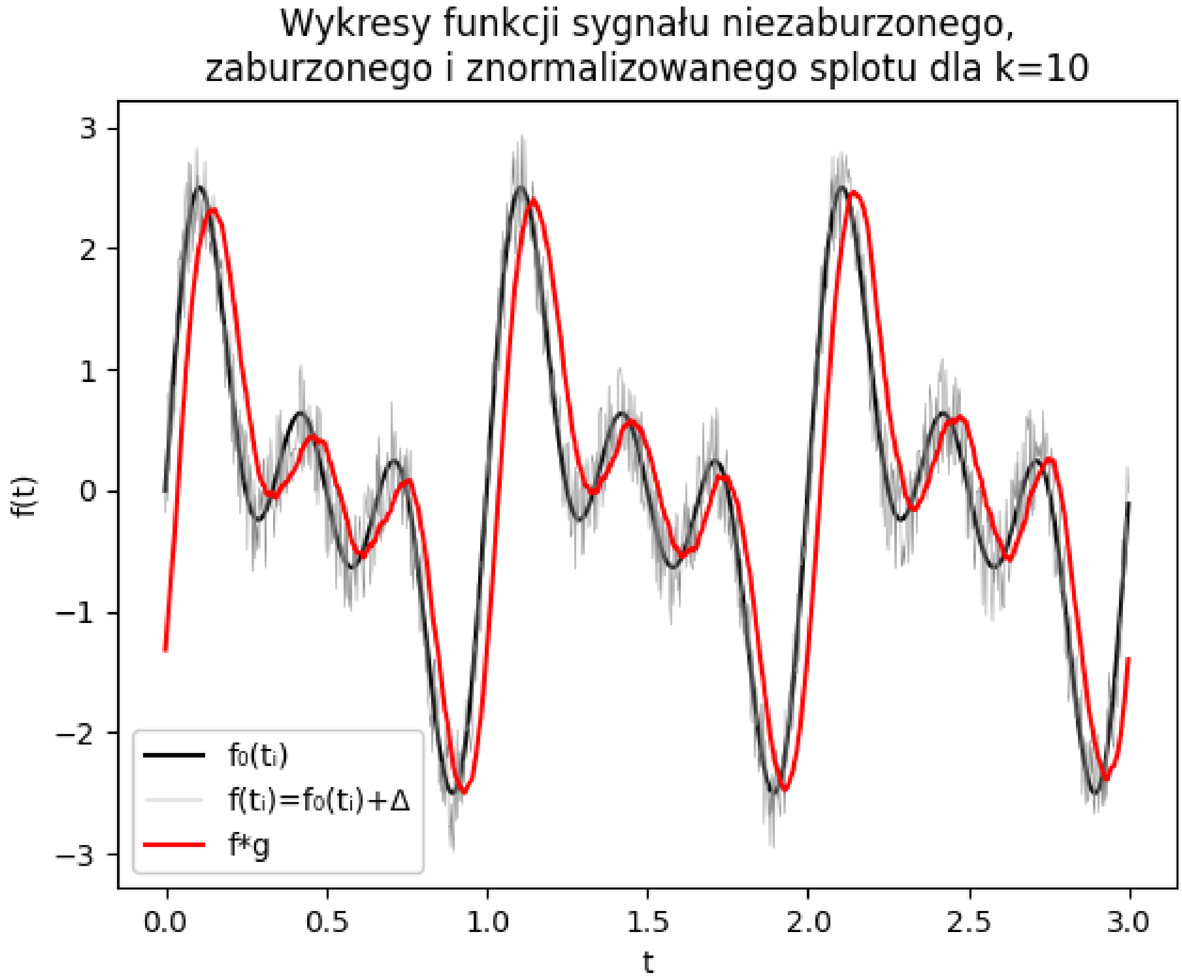
$$f * g \tag{16}$$

3 Wyniki

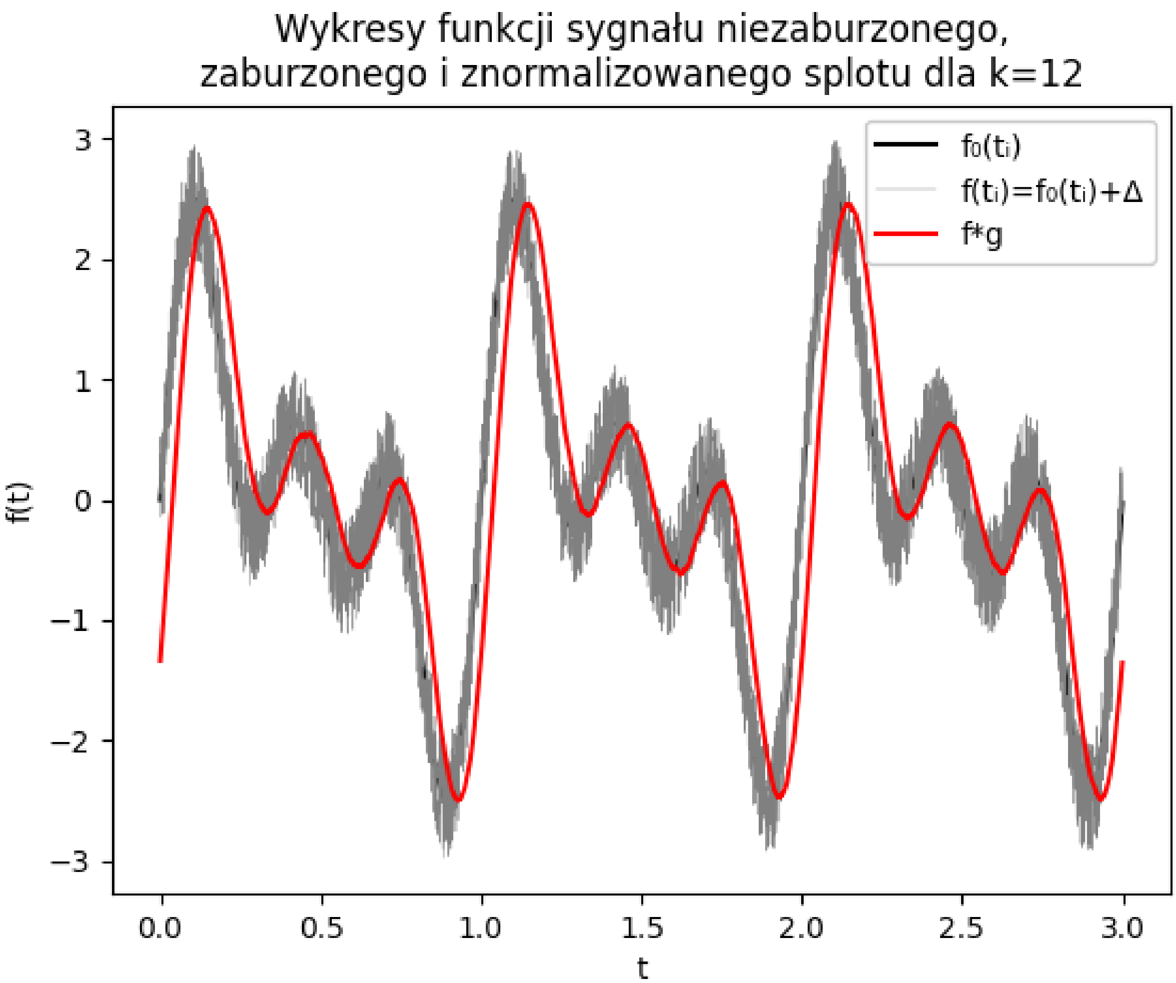
Cały program został napisany w języku Python. Za pomocą funkcji bibliotecznej matplotlib wyrysowaliśmy wykresy. Zostały one umieszczone poniżej na których widać, że wraz ze wzrostem liczby węzłów zwiększa się dokładność odszumionej funkcji. Transformaty Fouriera liczyliśmy za pomocą funkcji bibliotecznej fft() oraz ifft()



(a) Wykres dla funkcji $f_0(t) = sin(1 \cdot \omega t) + sin(2 \cdot \omega t) + sin(3 \cdot \omega t)$



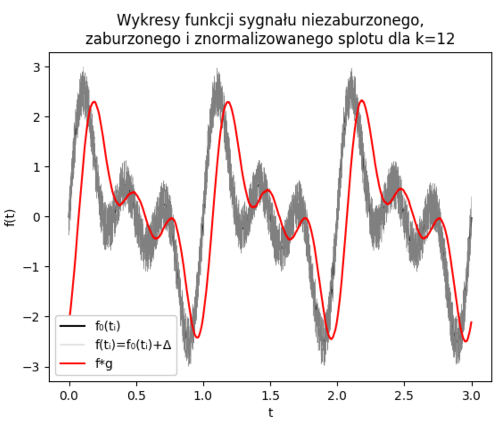
(a) Wykres dla funkcji $f_0(t) = sin(1 \cdot \omega t) + sin(2 \cdot \omega t) + sin(3 \cdot \omega t)$



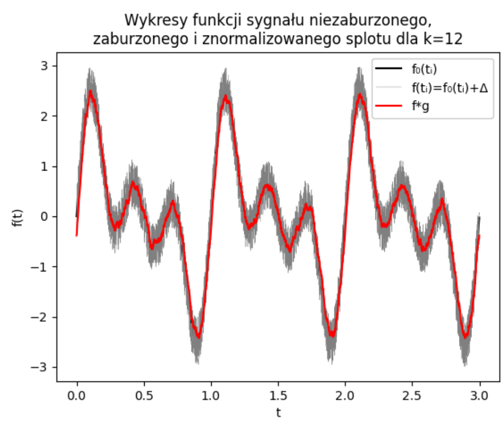
(a) Wykres dla funkcji $f_0(t) = sin(1 \cdot \omega t) + sin(2 \cdot \omega t) + sin(3 \cdot \omega t)$

4 Podsumowanie

Podsumowując można powiedzieć że Szybka Transformacja Fouriera pozwoliła na dokładne i szybkie przybliżenie funkcji zaszumionej. Dla większej liczby węzłów przybliżenie to było jeszcze bardziej dokładne. Szybka Transformacja Fouriera jest narzędziem stosowanym w wielu ważnych dziedzinach nauki i techniki. Jak zostało zaprezentowane powyżej, umożliwia między innymi szybkie i stosunkowo dokładne odszumianie sygnału wejściowego. Na podstawie zamieszczonych w sprawozdaniu rysunków, stwierdziliśmy, że zbyt mała ilość węzłów (zmniejszenie częstości próbkowania) indukuje słabą jakość odszumiania, wygładzania sygnału. Niestety w żadnym z przedstawionych przypadków, niezależnie od ilości węzłów, funkcja uzyskana w wyniku odszumienia nie stanowiła "całkowicie" wiernego przybliżenia funkcji rzeczywistej (niezaszumionego sygnału) - nie dla każdego momentu czasu wartości tych funkcji były sobie równe.
Przyczyny takiego stanu rzeczy można szukać w parametrze (odchyleniu standardowym) ze wzoru. Jest on stanowczo za wielki, zmiana jego wartości na mniejszą skutkuje lepszą jakością wyników.



(a) Wykres dla $\sigma = \frac{T}{10}$



(b) Wykres dla $\sigma = \frac{T}{100}$

Rysunek 4: Porównanie działania programu dla różnych wartości σ Wykresy dla funkcji $f_0(t) = sin(1 \cdot \omega t) + sin(2 \cdot \omega t) + sin(3 \cdot \omega t)$

Dzięki zastosowaniu w naszym programie algorytmu bazującego na metodzie "dziel i zwyciężaj" (tj. fft), która ogranicza czas obliczeń do $O(nlog(n))$ możemy powiedzieć iż jest to dośc szybka metoda.
Jak można zauważyć z wykresu 1 odszumiona funkcja już dla najmniejszego sprawdzanego k "wmiare" dokładnie odzwierciedla funkcję pierwotną, przez co zwiększenie k nie zwiększa diametralnie dokładności pomiędzy k = 10, a k = 12.
Biorąc pod uwagę dokładność otrzymanych wyników oraz krótki czas obliczeń można wysnuć wniosek o skuteczności wybranej metody w rozwiązywaniu postawionego problemu.

5 Literatura

[1] Tomasz Chwiej, Szybka transformacja Fouriera
http://home.agh.edu.pl/ chwiej/mn/fft_1819.pdf

[2] Wikipedia, Rozkład normalny
https://pl.wikipedia.org/wiki/Rozk%C5%82ad_normalny

[2] Wikipedia, Transformacja Fouriera
https://pl.wikipedia.org/wiki/Transformacja_Fouriera