

Metody Obliczeniowe w Nauce i Technice

16. Szeregi i transformaty Fouriera

Marian Bubak, Katarzyna Rycerz

Department of Computer Science
AGH University of Science and Technology
Krakow, Poland
kzajac@agh.edu.pl
dice.cyfronet.pl

Contributors

Maciej Trzebiński
Mikołaj Biel
Rafał Stachura



Outline

- 1 Wstęp
- 2 Podstawowe własności szeregów i transformat Fouriera
- 3 Interpolacja trygonometryczna
- 4 Szybka transformata Fouriera FFT dla $n = 2^m$
- 5 Transformata Hartley'a
- 6 Transformata Fouriera w internecie

Wstęp

Przykładowe zastosowania transformaty Fouriera:

- a) metody spektralne:
 - fizyka, chemia: badanie właściwości atomów, cząsteczek, itp.
 - na podstawie widma promieniowania elektromagnetycznego
- b) algorytmy numeryczne:
 - równania różniczkowe
 - analiza: badanie jakości algorytmów (np. dla MES)
- c) cyfrowe przetwarzanie sygnału
 - badanie składowych harmoniczných
 - filtracja obrazów i dźwięku
 - kompresja

Sposoby opisu procesu fizycznego I

① w dziedzinie czasu (time domain) $\implies A(t)$
 t – czas, A – pewna wielkość

② w dziedzinie: $\left. \begin{array}{l} \text{częstości}(\omega) \\ \text{częstotliwości}(f) \end{array} \right\}$

$$\implies \hat{A}(\omega) \quad \omega = \frac{2\pi}{T} = 2\pi f \text{ gdzie } f = \frac{1}{T}$$

$A(t)$, $\hat{A}(\omega)$ – dwie różne reprezentacje tego samego zjawiska związane równaniami transformat Fouriera:

$$\left. \begin{aligned} A(t) &= \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \hat{A}(\omega) \cdot e^{i\omega t} \\ \hat{A}(\omega) &= \int_{-\infty}^{\infty} dt \cdot A(t) \cdot e^{-i\omega t} \end{aligned} \right\}$$

lub równoważnie:

Sposoby opisu procesu fizycznego II

$$\left. \begin{aligned} A(t) &= \int_{-\infty}^{\infty} df \cdot \hat{A}(f) \cdot e^{2\pi i f t} \\ \hat{A}(f) &= \int_{-\infty}^{\infty} dt \cdot A(t) \cdot e^{-2\pi i f t} \end{aligned} \right\} \rightarrow \begin{aligned} &\text{korzystamy z: } \omega = 2\pi f \\ &\text{nie trzeba pamiętać} \\ &\text{o czynniku } \frac{1}{2\pi} \end{aligned}$$

Trasformować można zarówno w czasie, jak i w przestrzeni, jeśli mamy:

t – czas $\rightleftharpoons \omega$ – częstość kołowa

x – położenie $\rightleftharpoons k = \frac{2\pi}{\lambda}$ – liczba falowa, λ – długość fali

$A(t)$, $\hat{A}(\omega)$ – ciągłe f. swych argumentów

$$A(t) \rightleftharpoons \hat{A}(\omega)$$

Definicje transformat Fouriera I

a) FT - transformata Fouriera (Fourier Transform)

x - ciągłe

k - ciągłe

$A(x), \hat{A}(k)$ - f. ciągłe



$$\begin{aligned}\hat{A}(k) &= \int_{-\infty}^{\infty} dx A(x) e^{-ikx} \\ A(x) &= \int_{-\infty}^{\infty} \frac{dk}{2\pi} \hat{A}(k) e^{ikx}\end{aligned}\tag{16.1}$$

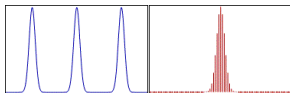
Definicje transformat Fouriera II

- ⓑ) W transformacie odwrotnej stosujemy - szereg Fouriera (Fourier Series)

x - zmienna ciągła

$B(x)$ - f. okresowa ciągłej zmiennej x ; okres: L

(ciągła odcinkami wraz z pochodną: na tych odcinkach - szereg zbieżny do $B(x)$, w punktach nieciągłości - do wartości średniej)



l - liczba całkowita

k - dyskretne

$$\begin{aligned}\hat{B}(k) &= \int_L dx B(x) e^{-ikx} \\ B(x) &= \frac{1}{L} \sum_{l=-\infty}^{\infty} \hat{B}(k) e^{ikx}\end{aligned}\quad (16.2)$$

$$k = \underbrace{\frac{2\pi}{L}}_{k_0} \cdot l = k_0 \cdot l$$

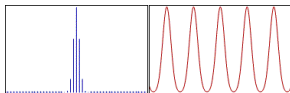
Definicje transformat Fouriera III

e) Discrete-time Fourier transform

x_p - dyskretne o skoku H

$$x_p = p \cdot H$$

p - liczba całkowita



k - ciągła

$\hat{C}(k)$ - periodyczna

okres $k_g = \frac{2\pi}{H}$

$$\begin{aligned}\hat{C}(k) &= H \cdot \sum_{p=-\infty}^{\infty} C(x_p) e^{-ikx_p} \\ C(x_p) &= \int_{k_g} \frac{dk}{2\pi} \hat{C}(k) e^{ikx_p}\end{aligned}\tag{16.3}$$

Definicje transformat Fouriera IV

d) fFT - skończona transformata Fouriera (finite FT)

x_p - dyskretne o skoku H

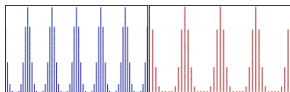
$x_p = p \cdot H$

$D(x_p)$ - okresowa; okres: L

N - ilość punktów w okresie $D(x_p)$

k - dyskretna, skok $k_0 = \frac{2\pi}{L}$; $k = l \cdot k_0$

$\hat{D}(k)$ - okresowa, okres $k_g = \frac{2\pi}{H}$

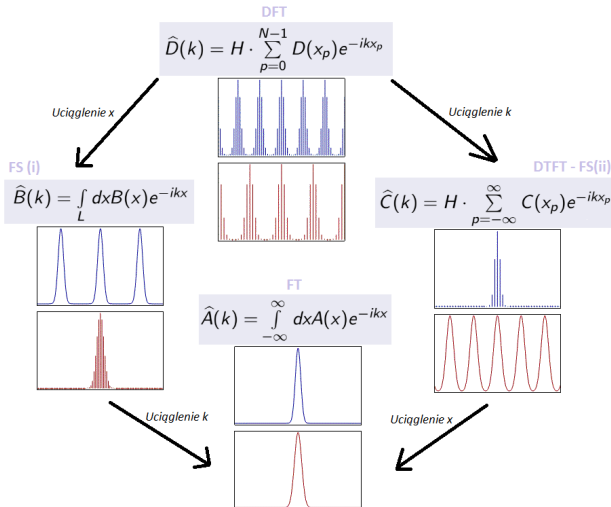


$$\hat{D}(k) = H \cdot \sum_{p=0}^{N-1} D(x_p) e^{-ikx_p} \quad (16.4)$$

$$D(x_p) = \frac{1}{L} \sum_{l=0}^{N-1} \hat{D}(k) e^{ikx_p}$$

Związki między transformatami Fouriera

Przez przejścia graniczne:



Zestawienie własności transformat I

poza własnościami dotyczącymi pochodnej - obowiązują dla wszystkich 4 transformat

$$Z : f(x) \rightleftharpoons \hat{f}(k); \quad g(x) \rightleftharpoons \hat{g}(k)$$

podobieństwo: $f\left(\frac{x}{a}\right) \rightleftharpoons |a| \cdot \hat{f}(k \cdot a)$

mnożenie przez stałą: $b \cdot f \rightleftharpoons b \cdot \hat{f}$

suma: $f + g \rightleftharpoons \hat{f} + \hat{g}$

odwrotność: jeżeli: $f(x) \rightleftharpoons \hat{f}(k) = g(k)$ (16.7)

to: $g(x) \rightleftharpoons \hat{g}(k) = 2\pi \cdot f(-k)$

przesunięcie: $f(x + a) \rightleftharpoons e^{ika} \cdot \hat{f}(k)$

pochodna: $\frac{df}{dx} \rightleftharpoons ik \cdot \hat{f}(k)$

Zestawienie własności transformat II

Twierdzenie o mocy

$$\int_{-\infty}^{\infty} f(x) \cdot g^*(x) dx = \int_{-\infty}^{\infty} \hat{f}(k) \cdot \hat{g}^*(k) \frac{dk}{2\pi} \quad (16.8)$$

Transformata zachowuje iloczyn skalarny.

Jeśli ta sama funkcja \rightarrow kwadrat modułu \rightarrow moc sygnału

Analogicznie dla FS i fFT (wtedy sumy szeregów)

Splot (konwolucja)

$f(x), g(x)$ – funkcje
ich splot:

$$h(x) = \int_{-\infty}^{\infty} dx' f(x') g(x - x') \equiv f * g \quad (16.9)$$

Własności:

$f * g = g * f$	przemienny
$f * (g * h) = (f * g) * h$	łączny
$f * (g + h) = f * g + f * h$	rozdzielny

Przykładowe zastosowanie: splot sygnału z filtrem

Przykład operacji splotu dla dwóch wektorów kolumnowych:

$$a \in \mathbb{R}^{n \times 1} ; b \in \mathbb{R}^{n \times 1} ; a = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} ; b = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{bmatrix}$$

$$[a * b]_k = \sum_{i=0}^k \alpha_i \cdot \beta_{k-i} ; a * b \in \mathbb{R}^{2n \times 1}$$

$$\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix} \begin{matrix} \downarrow \\ \begin{bmatrix} \beta_{n-1} \\ \vdots \\ \beta_1 \\ \beta_0 \end{bmatrix} \end{matrix} \Rightarrow a * b = \begin{bmatrix} \alpha_0 \beta_0 \\ \alpha_0 \beta_1 + \alpha_1 \beta_0 \\ \alpha_0 \beta_2 + \alpha_1 \beta_1 + \alpha_2 \beta_0 \\ \vdots \\ \alpha_{n-2} \beta_{n-1} + \alpha_{n-1} \beta_{n-2} \\ \alpha_{n-1} \beta_{n-1} \\ 0 \end{bmatrix}$$

Splot i jego transformaty

$$h(x) = f(x) * g(x) \leftrightarrow \widehat{h(k)} = \widehat{f(k)} \cdot \widehat{g(k)}$$

$$h(x) = f(x) \cdot g(x) \leftrightarrow \widehat{h(k)} = \widehat{f(k)} * \widehat{g(k)}$$

	x	k
FT	$\int_{-\infty}^{\infty} dx' \cdot f(x') \cdot g(x - x')$ $f(x) \cdot g(x)$	$\widehat{f(k)} \cdot \widehat{g(k)}$ $\int_{-\infty}^{\infty} \frac{dk'}{2\pi} \cdot \widehat{f(k')} \cdot \widehat{g(k - k')}$
FS(i)	$\int_L dx' \cdot f(x') \cdot g(x - x')$ $f(x) \cdot g(x)$	$\widehat{f(k)} \cdot \widehat{g(k)}$ $\frac{1}{L} \cdot \sum_{L'=-\infty}^{\infty} \widehat{f(k')} \cdot \widehat{g(k - k')}$
FS(ii)	$H \cdot \sum_{p'=-\infty}^{\infty} f(x'_p) \cdot g(x_p - x'_p)$ $f(x_p) \cdot g(x_p)$	$\widehat{f(k)} \cdot \widehat{g(k)}$ $\int_{k_g} \frac{dk'}{2\pi} \cdot \widehat{f(k')} \cdot \widehat{g(k - k')}$
fFT	$H \cdot \sum_{p'=0}^{\infty} f(x'_p) \cdot g(x_p - x'_p)$ $f(x_p) \cdot g(x_p)$	$\widehat{f(k)} \cdot \widehat{g(k)}$ $\frac{1}{l} \cdot \sum_{l'=0}^{N-1} \widehat{f(k')} \cdot \widehat{g(k - k')}$

Transformaty 3-D i ...

Uogólnienie z 1-D na 3-D (i więcej) - bezpośrednio:

1-D	3-D	
x	$\vec{x} = (x_1, x_2, x_3)$	
k	$\vec{k} = (k_1, k_2, k_3)$	
$k \cdot x$	$\vec{k} \cdot \vec{x}$	
dx	$d\vec{x}$	(16.11)
$\frac{d\vec{k}}{2\pi}$	$\frac{d\vec{k}}{(2\pi)^3}$	
L	$V_b = L_1 \cdot L_2 \cdot L_3$	
H	$V_c = H_1 \cdot H_2 \cdot H_3$	
	...	

Interpolacja trygonometryczna

- Wielomiany algebraiczne nie są dobre do opisu zjawisk okresowych
- Rozwiązanie: interpolacja wielomianami opartymi o funkcje trygonometryczne

funkcje okresowe o okresie L spełniają zależność $g(y + L) = g(y)$
funkcje trygonometryczne: okresem jest 2π , po przeskalowaniu:

$$x = \frac{2\pi}{L} \cdot y, f(x) = g\left(\frac{x \cdot L}{2\pi}\right)$$

okresem $g(y)$ jest L , okresem $f(x)$ jest 2π ,

$$f(x + 2\pi) = g\left(\frac{(x + 2\pi) \cdot L}{2\pi}\right) = g\left(\frac{x \cdot L}{2\pi} + L\right) = g\left(\frac{x \cdot L}{2\pi}\right) = f(x)$$

Interpolacja trygonometryczna

szukamy wielomianu trygonometrycznego

$$t_{n-1}(x) = \sum_{j=0}^{n-1} c_j \cdot (e^{ix})^j = \sum_{j=0}^{n-1} c_j \cdot e^{ijx}$$

gdzie (przypomnienie - wzór Eulera)

$$e^{ix} = \cos(x) + i\sin(x)$$

który w n punktach $x_k \in (0, 2\pi]$ przyjmuje te same wartości, co interpolowana funkcja

$$t_{n-1}(x_k) = f(x_k), k = 0, 1, \dots, n-1$$

Interpolacja trygonometryczna

Wielomiany trygonometryczne różnią się od algebraicznych tylko wyborem zmiennej:

Theorem

Zadanie interpolacji trygonometrycznej ma jednoznaczne rozwiązanie.

W praktyce - ważny przypadek szczególny:

n węzłów równoodległych, $x_k = \frac{2\pi}{n} \cdot k$, $k = 0, 1, \dots, n-1$

funkcje e^{ijx} , $j = 0, 1, \dots, n-1$ tworzą układ ortogonalny w sensie iloczynu skalarnego zdefiniowanego:

$$\langle f | g \rangle = \sum_{k=0}^{n-1} f(x_k) \cdot g^*(x_k), x_k = \frac{2\pi}{n} \cdot k, k = 0, 1, \dots, n-1$$

Interpolacja trygonometryczna

A dokładniej:

$$\langle e^{ijx} | e^{ilx} \rangle = \sum_{k=0}^{n-1} e^{ijx_k} \cdot e^{-ilx_k} = n \cdot \delta_{j,l} = \begin{cases} 0 & j \neq l \\ n & j = l \end{cases}$$

Dla

$$x_k = \frac{2\pi}{n} \cdot k, k = 0, 1, \dots, n-1$$

Gdzie $\delta_{j,l}$ to delta Kroneckera

$$\delta_{j,l} = \begin{cases} 0, & j \neq l \\ 1, & j = l \end{cases} \quad \text{dla } j, l \in \{0, 1, \dots, n-1\}$$

Interpolacja trygonometryczna

Dowód:

$$\langle e^{ijx} | e^{ilx} \rangle = \sum_{k=0}^{n-1} e^{ijx_k} \cdot e^{-ilx_k} = \sum_{k=0}^{n-1} e^{i(j-l)\frac{2\pi k}{n}} (*)$$

dla $j = l$

$$(*) = \sum_{k=0}^{n-1} e^0 = n$$

dla $j \neq l$

$$(*) = \sum_{k=0}^{n-1} \underbrace{e^{\frac{i(j-l)2\pi}{n} \cdot k}}_{\substack{\text{-szereg geom.} \\ \text{-n-wyrazów}}} = a_0 \frac{1 - q^n}{1 - q} = \frac{1 - \overbrace{e^{\frac{i(j-l)2\pi}{n} \cdot n}}^{\cos((j-l)2\pi) + i\sin((j-l)2\pi) = 1}}{1 - e^{\frac{i(j-l)2\pi}{n}}} = 0$$

$$a_0 = 1; q = e^{\frac{i(j-l)2\pi}{n}}$$

Interpolacja trygonometryczna

Jakie powinny być współczynniki wielomianu interpolacyjnego c_j ?

1°

$$\begin{aligned}\langle t_{n-1}(x) | e^{ilx} \rangle &= \left\langle \sum_{j=0}^{n-1} c_j \cdot e^{ijx} \middle| e^{ilx} \right\rangle = \sum_{j=0}^{n-1} c_j \cdot n \cdot \delta_{jl} = \\ &= c_l \cdot n \implies \underline{c_l = \frac{1}{n} \langle t_{n-1}(x) | e^{ilx} \rangle}\end{aligned}$$

2°

$$\langle t_{n-1}(x) | e^{ilx} \rangle = \sum_{k=0}^{n-1} \underbrace{t_{n-1}(x_k)}_{\substack{= \text{wartości} \\ \text{funkcji}}} \cdot e^{-ilx_k} = \sum_{k=0}^{n-1} f(x_k) \cdot e^{-ilx_k}$$

Interpolacja trygonometryczna

czyli:

$$c_j = \frac{1}{n} \sum_{k=0}^{n-1} f(x_k) \cdot e^{-ijx_k}, j = 0, 1, \dots, n-1$$

Dwa etapy obliczeń:

analiza Fouriera:

dla danych liczb zespolonych $f(x_k)$, $k = 0, 1, \dots, n-1$ szukamy $c_j, j = 0, 1, \dots, n-1$

synteza Fouriera:

mając liczby $c_j, j = 0, 1, \dots, n-1$ szukamy

$$f(x) = \sum_{j=0}^{n-1} c_j \cdot e^{ij\frac{2\pi}{n} \cdot k}, k = 0, 1, \dots, n-1$$

Interpolacja trygonometryczna

\implies obie:

- dyskretne
- wzajemnie odwrotne

Podsumowanie:

klasyczny algorytm: ($f = A \cdot C$)

$$n^2 \begin{cases} \text{-zespólonych mnożeń,} \\ \text{-zespólonych dodawań,} \\ \text{-obliczeń } e^{-i\frac{2\pi kj}{n}} \end{cases}$$

Wada \rightarrow duża złożoność operacji

FFT I

- Danielson, Lanczos (1942)
- R.L.Garwin (IBM Yorktown Heights Research Center)
- James William Cooley, John Wilder Tukey (1962)

Table: Złożoność

	obliczeniowa		czasowa
klasyczna FT:	$O(N^2)$	\Rightarrow	1.5 godziny
FFT:	$O(N \log_2 N)$	\Rightarrow	0.1 sekundy

Założenia:

rozmiar problemu: $N = 10^6$

CPU ~ 100 MFLOPS

FFT II

Dane: $f(x_k), x_k = \frac{2\pi}{n} \cdot k, k = 0, 1, \dots, n-1$

Szukamy: $c_j = \frac{1}{n} \sum_{k=0}^{n-1} f(x_k) \cdot e^{-i\frac{2\pi k}{n} \cdot j}, j = 0, 1, \dots, n-1$

gdy: $a_k = \frac{1}{n} \cdot f(x_k), \omega = e^{-i\frac{2\pi}{n}}$

to: $c_j = \sum_{k=0}^{n-1} a_k \cdot \omega^{jk}, j = 0, 1, \dots, n-1$

Założenie: ilość punktów: $n = 2^m \implies$ tyleż współczynników Fouriera.

FFT III

(k - numer punktu)

gdy k parzyste $k = 2 \cdot k_1$

gdy k nieparzyste $k = 2 \cdot k_1 + 1$

Dziedzina k:

z dołu $\underline{k_1 = 0}$ (parzyste k)

z góry: $n - 1 = 2^m - 1 \implies k$ - nieparzyste:

$$2k_1 + 1 = n - 1 \implies k_1 = \frac{n}{2} - 1$$

Rozdzielamy wyznaczanie współczynników!!!:

$$c_j = \sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1} \cdot (\omega^2)^{j \cdot k_1} + \sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1+1} \cdot (\omega^2)^{j \cdot k_1} \cdot \omega^j \quad (16.24)$$

FFT IV

$$\begin{cases} 0 \leq k_1 \leq \frac{n}{2} - 1 \\ 0 \leq j \leq n - 1 \end{cases}$$

Dzielimy liczby j na dwa zbiory wg zasady:

$$j = \frac{n}{2} \cdot l + j_1 \text{ dla (reszta z dzielenia) } 0 \leq j_1 \leq \frac{n}{2} - 1$$

Mniejsze od $\frac{n}{2}$

$$l = 0 \text{ dla } 0 \leq j \leq \frac{n}{2} - 1$$

Niemniejsze od $\frac{n}{2}$

$$l = 1 \text{ dla } \frac{n}{2} \leq j \leq n - 1$$

FFT V

z kolei:

$$\underline{(\omega^2)^{jk_1}} = \omega^{2 \cdot [\frac{n}{2} \cdot l + j_1] \cdot k_1} = \omega^{n \cdot l \cdot k_1 + 2j_1 k_1} = (e^{-\frac{2\pi i}{n}})^{(n \cdot l \cdot k_1 + 2j_1 k_1)} = \underline{(\omega^2)^{j_1 k_1}}$$

(bo $e^{-2\pi \cdot i \cdot l \cdot k_1} = 1$)

oraz

$$\omega^j = \omega^{\frac{n}{2} \cdot l + j_1} = (e^{-\frac{2\pi i}{n}})^{\frac{n}{2} \cdot l} \cdot \omega^{j_1} = e^{-i\pi l} \cdot \omega^{j_1}$$

Czyli

$$\begin{cases} \omega^j = \omega^{j_1} & \text{dla } l = 0 \text{ czyli } 0 \leq j \leq \frac{n}{2} - 1 \\ \omega^j = -\omega^{j_1} & \text{dla } l = 1 \text{ czyli } \frac{n}{2} \leq j \leq n - 1 \end{cases}$$

FFT VI

i uzyskujemy: Dla $0 \leq j \leq \frac{n}{2} - 1$

$$c_j = \underbrace{\sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1} \cdot (\omega^2)^{j_1 k_1}}_{\varphi(j_1)} + \underbrace{\sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1+1} \cdot (\omega^2)^{j_1 k_1} \cdot \omega^{j_1}}_{\psi(j_1)}, 0 \leq j_1 \leq \frac{n}{2} - 1$$

Dla $\frac{n}{2} \leq j \leq n - 1$

$$c_j = \underbrace{\sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1} \cdot (\omega^2)^{j_1 k_1}}_{\varphi(j_1)} - \underbrace{\sum_{k_1=0}^{\frac{n}{2}-1} a_{2k_1+1} \cdot (\omega^2)^{j_1 k_1} \cdot \omega^{j_1}}_{\psi(j_1)}, 0 \leq j_1 \leq \frac{n}{2} - 1$$

FFT VII

Każdy z 2 członów jest transformatą Fouriera - zamiast pojedynczej transformaty w n punktach \rightarrow suma 2 transformat w $\frac{n}{2}$ punktach wykorzystywanych dwa razy

Dla $0 \leq j \leq \frac{n}{2} - 1$

$$c_j = \varphi(j_1) + \omega^{j_1} \cdot \psi(j_1); j_1 = 0, 1, \dots, \frac{n}{2} - 1$$

Dla $\frac{n}{2} \leq j \leq n - 1$

$$c_j = \varphi(j_1) - \omega^{j_1} \cdot \psi(j_1); j_1 = 0, 1, \dots, \frac{n}{2} - 1$$

itd ... \rightarrow dokonując dalszych podziałów.

złożoność obliczeniowa $\leq \underline{2 \cdot N \log_2 N}$

FFT VIII

Zasada dziel i zwyciężaj!

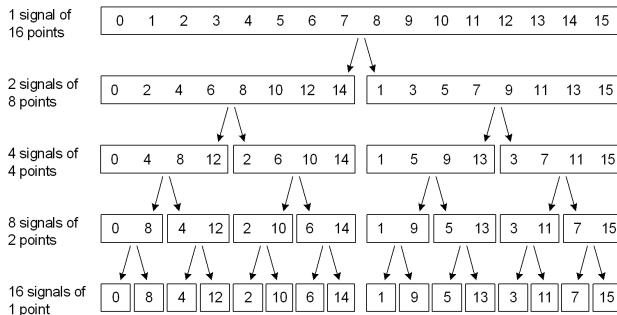


Figure: źródło: <https://riptutorial.com/algorithm/example/27088/radix-2-fft>

[//riptutorial.com/algorithm/example/27088/radix-2-fft](https://riptutorial.com/algorithm/example/27088/radix-2-fft)

FFT

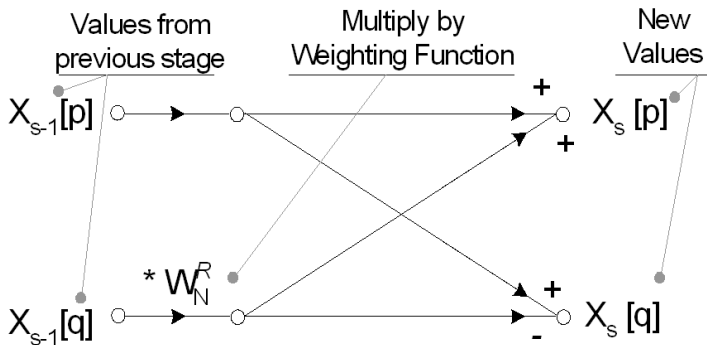


Figure: źródło: j.w.

$$W_N^R = (e^{\frac{-2\pi i}{N}})^R, \text{ W naszych oznaczeniach } N = n, R = j_1, W = \omega$$

FFT

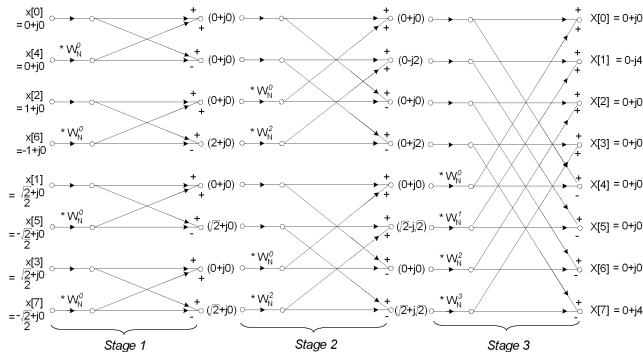


Figure: źródło: <https://riptutorial.com/algorithm/example/27088/radix-2-fft>

Rekurencyjny algorytm FFT

```
1 function FFT(a)
   $n \leftarrow \text{length}[a]$ 
3   if  $n = 1$ 
     then return a
5    $\omega_n \leftarrow e^{\frac{2\pi \cdot i}{n}}$ 
    $\omega \leftarrow 1$ 
7    $a_{\text{even}} \leftarrow (a_0, a_2, \dots, a_{n-2})$ 
    $a_{\text{odd}} \leftarrow (a_1, a_3, \dots, a_{n-1})$ 
9    $y^{\text{even}} \leftarrow \text{FFT}(a_{\text{even}})$ 
    $y^{\text{odd}} \leftarrow \text{FFT}(a_{\text{odd}})$ 
11  for  $j \leftarrow 0$  to  $\frac{n}{2} - 1$ 
      $y_j \leftarrow y_j^{\text{even}} + \omega y_j^{\text{odd}}$ 
13      $y_{j+\frac{n}{2}} \leftarrow y_j^{\text{even}} - \omega y_j^{\text{odd}}$ 
      $\omega \leftarrow \omega \cdot \omega_n$ 
15  end
   return y
17 end
```

Zastosowanie - szybkie mnożenie wielomianów

- Mamy dane $P(x) = \sum_{i=0}^{n-1} a_i x^i$ oraz $Q(x) = \sum_{i=0}^{n-1} b_i x^i$
- Jeśli potraktować $[a_i]$ oraz $[b_i]$ jak wektory, to $[c_i]$ gdzie $c_i = \sum_{j=0}^i a_j b_{i-j}$ jest ich splotem
- Jednocześnie: $W(x) = P(x) \cdot Q(x) = \sum_{i=0}^{2n-2} c_i x^i$ (aby wzór zadziałał, wyższe niż $n - 1$ współczynniki $P(x)$ oraz $Q(x)$ zastępujemy 0)
- zamiast wyliczać współczynniki wprost: transformata \rightarrow iloczyn \rightarrow odwrotna transformata

Splot współczynników wielomianów

$$(a_3x^3 + a_2x^2 + a_1x^1 + a_0) \cdot (b_3x^3 + b_2x^2 + b_1x + b_0) =$$

$$= \boxed{a_3b_3} x^6 + (\underline{a_2b_3} + \underline{a_3b_2}) x^5 + \dots + a_0b_0$$

$$c_6 x^6 = \left(\sum_{j=0}^6 a_j b_{6-j} \right) x^6 = (a_0 b_6 + a_1 b_5 + a_2 b_4 + \boxed{a_3 b_3}^{+0} + a_4 b_2 + a_5 b_1 + a_6 b_0) x^6$$

Zastosowanie - szybkie mnożenie wielomianów

Plan (szkic):

- 1 wyliczyć $2n-1$ wartości wielomianów $P(x_k)$ oraz $Q(x_k)$ dla $x_k = \omega^k = e^{\frac{2\pi i k}{2n-1}}$ dla $k = 0, \dots, 2n-2$ używając FFT (synteza) złożoność $O(n \log(n))$
- 2 policzyć wartości $W(x_k) = P(x_k) \cdot Q(x_k)$ - złożoność $O(n)$
- 3 policzyć współczynniki c_i znając wartości $W(x_k)$ - FFT (analiza) złożoność $O(n \log(n))$

Transformata Hartley'a I

Transformata Fouriera

$$F(f) = \int_{-\infty}^{\infty} X(t) e^{-i2\pi ft} dt$$

$$X(t) = \int_{-\infty}^{\infty} F(f) e^{i2\pi ft} df$$

$$c_j = \frac{1}{n} \sum_{k=0}^{n-1} X(t_k) e^{-i2\pi j \frac{k}{n}}$$

$$X(t_k) = \sum_{j=0}^{n-1} c_j e^{i2\pi j \frac{k}{n}}$$

Transformata Hartley'a II

Transformata Hartley'a

$$H(f) = \int_{-\infty}^{\infty} X(t) \operatorname{cas}(2\pi ft) dt$$

$$X(t) = \int_{-\infty}^{\infty} H(f) \operatorname{cas}(2\pi ft) dt$$

gdzie: $\operatorname{cas}(x) = \cos(x) + \sin(x)$

istotna różnica: zamiast $\underbrace{e^{-x \cdot i}}_{\text{zespólone}}$ mamy $\underbrace{\operatorname{cas}(x)}_{\text{rzeczywiste}}$ (\Rightarrow ilość

operacji arytmetycznych i pamięć)

Transformata Hartley'a III

Wersja dyskretna HT

$$H_j = \frac{1}{n} \sum_{k=0}^{n-1} f(t_k) \cdot \text{cas} \left(\frac{2\pi jk}{n} \right)$$

$$f(t_k) = \sum_{j=0}^{n-1} H_j \cdot \text{cas} \left(\frac{2\pi jk}{n} \right)$$

Bibliografia

- R.V.L. Hartley: A more symmetrical Fourier analysis applied to transmission problems, Proc. IRE, 30 (1942) 144,
- R.N. Bracewell: The fast Hartley transform, Proc. IEEE 72 (1984) 1010 (No 8),
- M.A. O'Neill: Faster than fast Fourier, Byte, April 1988, p.293.

FFT - przydatna w:

- ① analiza spektralna
- ② projektownie efektywnych algorytmów
 - iloczyn wielomianów \rightarrow spłot 2 wektorów
 - szybki binarny algorytm mnożenia liczb całkowitych (m. Schönhagego - Strassena)
- ③ \rightarrow A.V. Aho, J.E. Hopcroft, J.D. Ullman:
Projektowanie i analiza algorytmów komputerowych. PWN,
1983 (1974)

Transformata Fouriera w internecie

Biblioteki dla FFT w różnych językach:

- Transformata Fourier'a w ujęciu Matlab'a
<https://www.mathworks.com/help/matlab/fourier-analysis-and-filtering.html>
- Transformata Fourier'a w języku C (GSL) <https://www.gnu.org/software/gsl/doc/html/fft.html>
- FFTPACK5 - biblioteka dla języka Fortran: https://people.sc.fsu.edu/~jburkardt/f_src/fftpack5/fftpack5.f90
- NAG Library - The Numerical Algorithms Group FFT
https://www.nag.co.uk/numeric/fn/manual/pdf/c07/c07m01_fft_fn03.pdf

Więcej o zastosowaniach FT:

- Chemia i Medycyna:
 - spektroskopowe badanie właściwości materii:
<https://bit.ly/2rjpVXI>
 - Spektroskopia NMR:
Bernhard J., Analytische Chemie IV-Structure determination by NMR, 1. Practical aspects of pulse Fourier transform NMR spectroscopy, 1.1-1.21
 - FTIR (Fourier-transform infrared spectroscopy):
<https://bit.ly/2RsUEMS>

Transformata Fouriera w internecie

Informacje dodatkowe:

- Prosty poradnik wprowadzający do FT
<https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/>
- Zastosowania Transformaty Fourier'a(Stanford University)
<https://see.stanford.edu/materials/lsoftaee261/book-fall-07.pdf>