1 Szybka transformata Fouriera (FFT) funkcji rzeczywistej

Zdarza sie czesto, ze potrzebujemy transformaty Fouriera rzeczywistej funkcji. Wszystkie probki f_j, $j = 0, \cdots, N-1$ sa wtedy rzeczywiste. Oczywiscie mozemy uzyc procedury do FFT wstawiajac wartosci f_j do czesci rzeczywistej zespolonej macierzy wejsciowej kladac rownoczesnie wszystkie czesci urojone rowne zero. Otrzymana transformata F_n, $n = 0, \cdots, N-1$ spelnia $F^*_{N-n} = F_n$ ($H(-f) = [H(f)]^*$ gdy $h(t)$ jest rzeczywiste). Poniewaz zespolona macierz wyjsciowa ma rzeczywiste wartosci F_0 i $F_{N/2}$ oraz $(N/2)-1$ niezaleznych zespolonych wartosci $F_1, \cdots, F_{N/2-1}$ (pozostale wartosci sa do nich sprzezone) ma ona tyle samo $2(N/2-1)+2 = N$ stopni swobody co oryginalna, wejsciowa macierz. Mimo to uzycie zespolonego algorytmu FFT do funkcji rzeczywistej jest bardzo nieefektywne, tak w wykorzystaniu czasu jak i pamieci komputera.

Problem mozna rozwiazac ladniej poprzez podzielenie wejsciowego zbioru punktow f_j na polowy i tworzac dwa rzeczywiste zbiory dwukrotnie krotsze od poczatkowego. Wlasciwy podzial polega na wzieciu elementow parzystych jako pierwszy zbior a nieparzystych jako drugi. Stanowia one odpowiednio rzeczywista i zespolona czesc zespolonej macierzy h_j, $h_j = f_{2j} + if_{2j+1}$, $j = 0, \cdots, N/2-1$. Algorytm FFT daje nam zespolona macierz $H_n = F^e_n + iF^o_n$, $n = 0, \cdots, N/2-1$ z

$$\begin{aligned} F^e_n &= \sum_{k=0}^{N/2-1} f_{2k} e^{2\pi ikn/(N/2)} \\ F^o_n &= \sum_{k=0}^{N/2-1} f_{2k+1} e^{2\pi ikn/(N/2)}. \end{aligned} \tag{1}$$

Uwaga - oba F^e_n oraz F^o_n sa wielkosciami zespolonymi, nie stanowia one czesci rzeczywistej i urojonej H_n.

Pamietajac wyprowadzenie algorytmu FFT pamietamy, ze dla prawdziwej transformaty F_n funkcji f_j zachodzi

$$F_n = F^e_n + e^{2\pi in/N} F^o_n \quad n = 0, \cdots, N-1 \tag{2}$$

Stad otrzymujemy

$$F_n = \frac{1}{2}(H_n + H^*_{N/2-n}) - \frac{i}{2}(H_n - H^*_{N/2-n})e^{2\pi in/N} \quad n = 0, \cdots, N-1 \tag{3}$$

Problem w powyzszym schemacie stanowi rozseparowanie H_n na czesci F^e_n oraz F^o_n. Wykorzystuje sie do tego dwa fakty.

1. Dyskretna transformata funkcji rzeczywistej spelnia
$$F_{N-n} = (F_n)^* \tag{4}$$

2. Dyskretna transformata funkcji zespolonej spelnia
$$G_{N-n} = -(G_n)^* \tag{5}$$

W naszym problemie F^e_n to transformata funkcji reczywistej zas F^o_n funkcji zespolonej.

Symetrie te daja:

$$\begin{aligned} \frac{1}{2}(Re(H_k) + Re(H_{N-k})) &= Re(F^e_k) = Re(F^e_{N-k}) \\ \frac{1}{2}(Re(H_k) - Re(H_{N-k})) &= -Im(F^o_k) = Im(F^o_{N-k}) \\ \frac{1}{2}(Im(H_k) + Im(H_{N-k})) &= Re(F^o_k) = Re(F^o_{N-k} \\ \frac{1}{2}(Im(H_k) + Im(H_{N-k})) &= Im(F^e_k) = -Im(F^e_{N-k}) \end{aligned} \tag{6}$$

Kilka uwag:

- Poniewaz $F^*_{N-n} = F_n$ nie ma sensu zapamietywac calego spektrum. Dodatnia polowa spektrum jest wystarczajaca. Moze byc zapisana w tej samej macierzy co wejsciowe punkty.
- W procedurze potrzebujemy wartosci H_n, $n = 0, \cdots, N/2$ podczas gdy FFT daje nam tylko wartosci dla $n = 0, \cdots, N/2-1$. Ratuje nas symetria $H_{N/2} = H_0$.

2 Szybka transformata sinusowa i cosinusowa.

Poza innymi zastosowaniami transformata Fouriera moze sluzyc do rozwiazywania rownan rozniczkowych. Jak wiemy rownania moga miec dwa typy warunkow brzegowych

1. wartosc rozwiazania na brzegu rowna sie zero
2. wartosc pochodnej rozwiazania na brzegu rowna sie zero

2.1 Szybka transformata sinusowa

W pierwszym przypadku, naturalnym jest skorzystac z transformaty sinusowej, danej przez wzor

$$F_k = \sum_{j=1}^{N-1} f_j \sin(\pi jk/N) \tag{7}$$

gdzie f_j, $j = 0, \cdots, N-1$ to dane pukty a $f_0 = 0$.

W pierwszej chwili formula ta wydaje sie przedstawiac urojona czesc dyskretnej transformaty Fouriera (FFT)

$$F_k = \sum_{j=0}^{N-1} f_j e^{2\pi ijk/N} \tag{8}$$

Tak jednak nie jest. Argument sinusa w obu powyzszych wzorach rozni sie o czynnik dwa. Transformata sinusowa korzysta *wylacznie z sinusow* jako calkowitej bazy funkcji w zakresie od 0 do 2π. Transformata cosinusowa korzysta natomiast wylacznie z cosinusow. Porownujac, normalna szybka transformata uzywa obydwu sinusow i cosinusow, natomiast potrzebuje dwa razy mniej z kazdych z nich.

Wyrazenie (1) mozna "dopasowac" do postaci pozwalajacej na jego obliczenie przy wykorzystaniu FFT. Idea polega na powiekszeniu zbioru danych punktow f_j (czyli przedluzeniu funkcji f) poza ostatnia wartosc F_{N-1}. Dwukrotnie powiekszamy zbior punktow w ten sposob, aby uczynic go "nieparzystym" wokol wartosci $j = N$, z $f_N = 0$

$$f_{2N-j} = -f_j \quad j = 0, \cdots, N-1. \tag{9}$$

Rozwazmy FFT takiej powiekszonej funkcji

$$F_k = \sum_{j=0}^{2N-1} f_j e^{2\pi ijk/(2N)}. \tag{10}$$

Polowa powyzszej sumy od $j = N$ do $j = 2N-1$ moze byc przepisana z podstawieniem $j' = 2N - j$

$$\begin{aligned} \sum_{j=N}^{2N-1} f_j e^{2\pi ijk/(2N)} &= \sum_{j'=1}^{N} f_{2N-j'} e^{2\pi i(2N-j')k/(2N)} \\ &= -\sum_{j'=0}^{N-1} f_{j'} e^{-2\pi ij'k/(2N)}. \end{aligned} \tag{11}$$

Otrzymujemy

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} f_j \left[e^{2\pi ijk/(2N)} - e^{-2\pi ijk/(2N)} \right] \\ &= 2i \sum_{j=0}^{N-1} f_j \sin(\pi jk/N) \end{aligned} \tag{12}$$

czyli z dokladnoscia do czynnika 2π uzyskujemy transformate sinusowa wykorzystujac FFT rozszerzonej funkcji.

Wade powyzszej metody stanowi jej nieefektywnosc. Objawia sie ona tym, ze liczymy dwukrotnie za duzo wartosci. Transformacja FFT daje zero dla czesci rzeczywistej kazdego elementu transformacji F_k. Sa to zera do niczego nam nie potrzebne.

Zastosujmy, wiec inna metode. Korzystajac z wartosci f_j, $j = 1, \cdots, N-1$ skonstruujmy pomocnicza macierz y_j

$$\begin{aligned} y_0 &= 0 \\ y_j &= \sin(j\pi/N)(f_j + f_{N-j}) + \frac{1}{2}(f_j - f_{N-j}) \quad j = 1, \cdots, N-1 \end{aligned} \tag{13}$$

Macierz ta jest tego samego wymiaru co oryginalna. Zauwazmy, ze pierwsza czesc y_j jest symetryczna wzgledem $j = N/2$ natomiast druga jest antysymetryczna. Mozemy to wykorzystac tworzac dwie wielkosci

$$\begin{aligned} R_k &= \sum_{j=0}^{N-1} y_j \cos(2\pi jk/N) \\ &= \sum_{j=1}^{N-1} (f_j + f_{N-j}) \sin(j\pi/N) \cos(2\pi jk/N) \\ &= \sum_{j=0}^{N-1} 2 f_j \sin(j\pi/N) \cos(2\pi jk/N) \\ &= \sum_{j=0}^{N-1} f_j \left[\sin \frac{(2k+1)j\pi}{N} - \sin \frac{(2k+1)j\pi}{N} \right] \\ &= F_{2k+1} - F_{2k-1} \end{aligned} \tag{14}$$

$$\begin{aligned} I_k &= \sum_{j=0}^{N-1} y_j \sin(2\pi jk/N) \\ &= \sum_{j=1}^{N-1} (f_j - f_{N-j}) \frac{1}{2} \sin(2\pi jk/N) \\ &= \sum_{j=0}^{N-1} f_j \sin(2\pi jk/N) \\ &= F_{2k} \end{aligned} \tag{15}$$

gdzie skorzystalismy z parzystosci cosinusa i nieparzystosci sinusa w celu wyciecia odpowiednio czesci nieparzystych w (8) i parzystych w (9).

Ostatecznie F_k moze byc obliczone nastepujaco

$$F_{2k} = I_k \quad F_{2k+1} = F_{2k-1} + R_k \quad k = 0, \cdots, (N/2-1). \tag{16}$$

Parzyste F_{2k} otrzymujemy stad bezposrednio. Wyrazy nieparzyste wymagaja rekurencji, ktorej punkt startowy mozemy wyznaczyc kladac w rownaniu (10) $k = 0$ i korzystajac z wlasnosci $F_1 = F_{-1}$

$$F_1 = \frac{1}{2} R_0 \tag{17}$$

Zadziwiajaco odwrotna transformata sinusowa jest identyczna transformacie. Stosujac ja dwukrotnie odtwarza sie wejsciowe punkty pomnozone przez czynnik $N/2$.

2.2 Szybka transformata cosinusowa

W przypadku drugiego rodzaju warunku brzegowego, gdy pochodna rozwiazania na brzegu ma znikac, naturalnym jest stosowac transformate cosinusowa. Istnieje kilka sposobow definiowania transformaty. Roznia sie one sposobem rozszerzenia zbioru danych f_k do dwukrotnie wiekszego, parzystego zbioru oraz jego wielkoscia (moze to byc $2N-1, 2N$ lub inna liczba punktow). W praktyce tylko dwie z wielu mozliwosci sa uzyteczne, wiec ograniczymy sie do pokazania wylacznie ich.

Pierwsza postac transformaty cosinusowej uzywa $N+1$ punktow

$$F_k = \frac{1}{2}[f_0 + (-1)^k f_N] + \sum_{j=1}^{N-1} f_j \cos(\pi jk/N). \tag{18}$$

Rozszerzamy dany zbior punktow do zbioru parzystego wzgledem $j = N$, z

$$f_{2N-j} = f_j, \quad j = 0, \cdots, N-1. \tag{19}$$

Biorac ponownie rozszerzona funkcje (4) i wykonujac analogiczne kroki jak w (5) odnajduje sie, ze transformata Fouriera to dwa razy transformata cosinusowa.

Podobnie jak w przypadku sinusow i tutaj transformata moze byc policzona bez podwojnej niewydajnosci. W tym wypadku pomocnicza funkcja przyjmuje postac

$$y_j = \frac{1}{2}(f_j + f_{N-j}) - \sin(j\pi/N)(f_j - f_{N-j}) \quad j = 0, \cdots, N-1 \tag{20}$$

W miejsce (10) otrzymujemy tym razem zaleznosci

$$F_{2k} = R_k \quad F_{2k+1} = F_{2k-1} + I_k \quad k = 0, \cdots, (N/2-1) \tag{21}$$

z poczatkowa wartoscia zaleznosci rekurencyjnej

$$F_1 = \frac{1}{2}(f_0 - f_N) + \sum_{j=1}^{N-1} f_j \cos(j\pi/N). \tag{22}$$

Ponownie transformata stanowi swa wlasna transformate odwrotna.

Druga wazna postac transformaty cosinusowej zdefiniowana jest nastepujaco

$$F_k = \sum_{j=0}^{N-1} f_j \cos\frac{\pi k(j+\frac{1}{2})}{N} \tag{23}$$

z odwrotna transformata

$$f_j = \frac{2}{N}(\sum_{j=0}^{N-1} *) F_k \cos\frac{\pi k(j+\frac{1}{2})}{N} \tag{24}$$

gdzie * przy symbolu sumy oznacza, ze przed wyrazem z $k=0$ stoi wspolczynnik $\frac{1}{2}$. Powyzsza forma powstaje poprzez rozszerzenie zbioru danych punktow dla $j = 0, \cdots, N-1$ do $j = 0, \cdots, 2N-1$, tak ze nowy zbior jest parzysty wzgledem punktu $N - \frac{1}{2}$ i periodyczny.

Tym razem pomocnicza funkcja ma postac

$$y_j = \frac{1}{2}(f_j + f_{N-j-1}) - \sin\frac{\pi(j+\frac{1}{2})}{N}(f_j - f_{N-j-1}) \quad j = 0, \cdots, N-1 \tag{25}$$

oraz znajduje sie, ze

$$F_{2k} = \cos\frac{\pi k}{N} R_k - \sin\frac{\pi k}{N} I_k \tag{26}$$

$$F_{2k-1} = \sin\frac{\pi k}{N} R_k + \cos\frac{\pi k}{N} I_k + F_{2k+1} \tag{27}$$

Rownanie (21) daje

$$F_{N-1} = \frac{1}{2} R_{N/2} \tag{28}$$

Wyrazy parzyste znajduje sie wiec bezposrednio z (20), podczas gdy nieparzyste otrzymuje sie poprzez rekurencje (21) od $k = N/2 - 1$ w dol korzystajac z (22) by wystartowac.

Przy opisie skorzystalismy ze zrodla:

- Numerical Recepies

Pawel Jankowski i Jakub Narebski