1 introduction

ネットで FFT を調べてみると，バタフライ演算を使うという説明とその図解をよく見る気がします。しかし，もとの DFT の式から変形して FFT の式を導出しているようなサイトは見つかりませんでした (調査不足かもしれない)。

というわけで，自分なりに式変形して DFT の式から FFT(データ数が 2 の累乗数の場合) の式の導出をしてみたので，まとめとしてこの記事を書きます

ほかに参考文献らしいものがないので間違ってる可能性も十分にあります。

2 FFT とは

FFT とは Fast Fourier Transform の略で，離散フーリエ変換を計算機上で高速に計算するアルゴリズムのことを言います。離散フーリエ変換は本来信号処理分野で使われるものですが，どうやら多倍長演算の積の計算アルゴリズムなどにも使われてたりするようです。

3 離散フーリ変換 (Discrete Fourier Transfotm)

離散フーリエ変換はフーリエ変換を離散化したもので，周波数解析などでよくつかわれます。順変換は，虚数単位 i[*1]，データ数 N，ネイピア数 e，円周率 π を用いて

$$F(x) = \sum_{t=0}^{N-1} f(t) e^{-i\frac{2\pi t x}{N}} \tag{3.1}$$

と定義され，逆変換は

$$f(t) = \frac{1}{N} \sum_{x=0}^{N-1} F(x) e^{i\frac{2\pi x t}{N}} \tag{3.2}$$

と定義されます。

この式のまま愚直にプログラムを書くと，計算量は $O(N^2)$ となり，非常に遅いプログラムになってしまいます。現在，離散フーリエ変換は通信や音声解析など幅広い分野で利用されていますが，計算に時間がかかるのでは意味がありません。そこで，高速に離散フーリエ変換が計算できるアルゴリズムが意味を持ってくるわけです。

4 Cooley-Tukey 型アルゴリズム

今回は，いくつかあるらしい FFT のアルゴリズムのうち，Cooley-Tukey 型と言われるアルゴリズムに従って式を変形していきます。

[*1] 信号処理の分野では慣例から虚数単位に j を用いることがあるが，今回は i を用いる

まず，計算を簡略化するために離散フーリエ変換 (以下，DFT と書く) の式 (3.1)，(3.2) を 0 以上の整数 t, x, k を用いて

$$\begin{cases} F(t) = \displaystyle\sum_{x=0}^{N-1} f(x) w_N^{tx} \\ f(x) = \dfrac{1}{N} \displaystyle\sum_{t=0}^{N-1} F(t) w_N^{-xt} \end{cases} \qquad (0 \leq t, x < N = 2^k) \tag{4.1}$$

と書き換えます。ここで，

$$w_N^a = e^{-i\frac{2\pi a}{N}}$$

としました。このとき w_N は

$$\begin{aligned} w_N^{a+\frac{N}{2}} &= -w_N^a \\ w_N^{a+N} &= w_N^a \end{aligned} \tag{4.2}$$

という特徴を持ちます。

4.1　順変換

まず，データ数 N の順変換について考えます。式 (4.1) の順変換の式を変形すると

$$\begin{aligned} F(x) &= \sum_{t=0}^{\frac{N}{2}-1} \left[f(2t) w_N^{2t\cdot x} + f(2t+1) w_N^{(2t+1)\cdot x} \right] \\ \rightleftharpoons F(x) &= \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{tx} + w_N^x \sum_{x=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{tx} \end{aligned}$$

と書くことができ，x が $0 \leq x < \frac{N}{2}$ であるような整数だとすると $F(x)$ は

$$F(x) = \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{tx} + w_N^x \sum_{t=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{tx} \tag{4.3}$$

$$F(x + \frac{N}{2}) = \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{t\left(x+\frac{N}{2}\right)} + w_N^{x+\frac{N}{2}} \sum_{t=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{t\left(x+\frac{N}{2}\right)} \tag{4.4}$$

という 2 つの式で表すことができます。式 (4.4) は w_N と $w_{\frac{N}{2}}$ の性質 (4.2) を考えると

$$F(x + \frac{N}{2}) = \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{tx} - w_N^x \sum_{t=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{tx}$$

と書くことができるため，式 (4.3)，(4.4) はそれぞれ

$$\begin{cases} F(x) = \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{tx} + w_N^x \sum_{t=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{tx} \\ F(x+\frac{N}{2}) = \sum_{t=0}^{\frac{N}{2}-1} f(2t) w_{\frac{N}{2}}^{tx} - w_N^x \sum_{t=0}^{\frac{N}{2}-1} f(2t+1) w_{\frac{N}{2}}^{tx} \end{cases} \quad (0 \le x < \frac{N}{2}) \tag{4.5}$$

と書くことができます。ここで，関数 $f_{\frac{N}{2}}(x)$ を $C_2(a,b) = 2a + b$ を用いて

$$f_{\frac{N}{2}}(x + \frac{N}{2} b_0) = \sum_{t=0}^{\frac{N}{2}-1} f(C_2(t, b_0)) w_{\frac{N}{2}}^{tx} \qquad (b_0 \in \{0,1\}, \quad 0 \le x < \frac{N}{2}) \tag{4.6}$$

と定義します。この関数 $f_{\frac{N}{2}}(x)$ を用いることで，式 (4.5) は

$$\begin{cases} F(x) = f_{\frac{N}{2}}(x) + w_N^x f_{\frac{N}{2}}(x + \frac{N}{2}) \\ F(x+\frac{N}{2}) = f_{\frac{N}{2}}(x) - w_N^x f_{\frac{N}{2}}(x + \frac{N}{2}) \end{cases} \qquad (0 \le x < \frac{N}{2}) \tag{4.7}$$

と書き換えることができます。

式 (4.6) はデータ数 $\frac{N}{2}$ の DFT になっていて，式 (4.7) は式 (4.6) の和になっています。つまり，データ数 N の DFT はデータ数 $\frac{N}{2}$ の DFT に分割することができたと言えるでしょう。

さらに，$f_{\frac{N}{2}}(x)$ を分割することを考えます。式 (4.5) と同様に考えると，式 (4.6) は

$$\begin{cases} f_{\frac{N}{2}}(x + \frac{N}{2} b_0) = \sum_{t=0}^{\frac{N}{4}-1} f(2C_2(2t, b_0)) w_{\frac{N}{4}}^{tx} + w_{\frac{N}{2}}^x \sum_{t=0}^{\frac{N}{4}-1} f(2C_2(t, b_0) + 1) w_{\frac{N}{4}}^{tx} \\ f_{\frac{N}{2}}(x + \frac{N}{2} b_0 + \frac{N}{4}) = \sum_{t=0}^{\frac{N}{4}-1} f(2C_2(2t, b_0)) w_{\frac{N}{4}}^{tx} - w_{\frac{N}{2}}^x \sum_{t=0}^{\frac{N}{4}-1} f(2C_2(t, b_0) + 1) w_{\frac{N}{4}}^{tx} \end{cases} \quad (b_0 \in \{0,1\}, \quad 0 \le x < \frac{N}{4}) \tag{4.8}$$

と書くことができ，$f_{\frac{N}{4}}(x)$ を

$$f_{\frac{N}{4}}(x + \frac{N}{2} b_0 + \frac{N}{4} b_1) = \sum_{t=0}^{\frac{N}{4}-1} f(C_2(C_2(t, b_1), b_0)) w_{\frac{N}{4}}^{tx} \qquad (b_0, b_1 \in \{0,1\}, \quad 0 \le x < \frac{N}{4}) \tag{4.9}$$

とすれば，(4.8) は

$$\begin{cases} f_{\frac{N}{2}}(x + \frac{N}{2} b_0) = f_{\frac{N}{4}}(x + \frac{N}{2} b_0) + w_{\frac{N}{2}}^x f_{\frac{N}{4}}(x + \frac{N}{2} b_0 + \frac{N}{4}) \\ f_{\frac{N}{2}}(x + \frac{N}{2} b_0 + \frac{N}{4}) = f_{\frac{N}{4}}(x + \frac{N}{2} b_0) - w_{\frac{N}{2}}^x f_{\frac{N}{4}}(x + \frac{N}{2} b_0 + \frac{N}{4}) \end{cases} \qquad (0 \le x < \frac{N}{2}) \tag{4.10}$$

と書くことができます。以上のように，データ数 N の DFT の順変換によって得られる $F(x)$ は $f_{\frac{N}{2}}(x)$ によって表すことができ，$f_{\frac{N}{2}}(x)$ は $f_{\frac{N}{4}}(x)$ によって表すことができることがわかりました。

以上の式から，データ数 N の DFT の順変換によって得られる $F(x)$ は式 (4.10)，(4.9) から，$0 < l < k$ に対して，

$$\begin{cases} F(x) = f_{\frac{N}{2^1}}(x) + w_N^x f_{\frac{N}{2^1}}(x + \dfrac{N}{2}) \\ F(x + \dfrac{N}{2}) = f_{\frac{N}{2^1}}(x) - w_N^x f_{\frac{N}{2^1}}(x + \dfrac{N}{2}) \end{cases} \qquad (0 \leq x < \frac{N}{2}) \tag{4.11}$$

$$\begin{cases} f_{\frac{N}{2^l}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1})) = f_{\frac{N}{2^{l+1}}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1})) + w^x_{\frac{N}{2^l}} f_{\frac{N}{2^{l+1}}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1} + \\ f_{\frac{N}{2^l}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1} + 1)) = f_{\frac{N}{2^{l+1}}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1})) - w^x_{\frac{N}{2^l}} f_{\frac{N}{2^{l+1}}}(x + \dfrac{N}{2^{l+1}}(2^l b_0 + ... + 2b_{l-1} + \end{cases} \tag{4.12}$$

であり，$l = k$ に対しては

$$f_{\frac{N}{2^k}}(2^k b_0 + ... + b_k) = f(C_2(...C_2(b_k, b_{k-1})..., b_0)) \tag{4.13}$$

と書くことができる考えられます。このとき，$b_0, b_1, ..., b_k \in \{0, 1\}$ です。

式 (4.11)，(4.12)，(4.13) が正しいかどうかを帰納法によって確認していきます。まず，$N = 2^1$ の場合を考えます。このとき，DFT の式 (4.1) は

$$\begin{aligned} F(x) &= \sum_{t=0}^{1} f(t) w_2^{tx} \\ \rightleftharpoons F(x) &= f(0) + w_2^x f(1) \qquad (x, t = 0, 1) \end{aligned}$$

となります。式 (4.13) から $f_{\frac{N}{2^1}}$ は

$$f_{\frac{N}{2^1}}(b_0) = f(b_0)$$

となると考えられます。さらに，式 (4.11) から $F(x)$ は

$$\begin{cases} F(0) = f_{\frac{N}{2^1}}(0) + w^x_{\frac{N}{2^1}} f_{\frac{N}{2^1}}(1) \\ F(1) = f_{\frac{N}{2^1}}(0) - w^x_{\frac{N}{2^1}} f_{\frac{N}{2^1}}(1) \end{cases}$$

となり，$N = 2^1$ のときは仮説と DFT の式が一致することが確認できました。

次に，$N = 2^k$ のときにも成立すると仮定して $N' = 2^{k+1}$ の場合を考えます。データ数 N' の DFT は式 (4.1) より

$$F(x) = \sum_{t=0}^{N'-1} f(t) w_{N'}^{tx}$$

と書けます。この式を分割すると

$$\begin{aligned} F(x) &= \sum_{t=0}^{N-1} f(2t) w_N^{tx} + w_{N'}^x \sum_{x=0}^{N-1} f(2t+1) w_N^{tx} \\ F(x + N) &= \sum_{t=0}^{N-1} f(2t) w_N^{tx} - w_{N'}^x \sum_{x=0}^{N-1} f(2t+1) w_N^{tx} \end{aligned}$$

となります。ここで，$f'(t) = f(2t)$，$f''(t) = f(2t+1)$ とし，$f'(t)$ を DFT したものを $f'_N(x)$，$f''(t)$ を DFT したものを $f''_N(x)$ とすれば，$F(x)$ をデータ数 N の DFT の和として書くことができます。データ数 N の DFT を式 (4.11) によって書き換えると

$$
\begin{aligned}
F(x) &= f'_N(x) + w^x_{N'} f''_N(x) \\
F(x+N) &= f'_N(x) - w^x_{N'} f''_N(x)
\end{aligned}
$$

とできます。このとき，$f'_N(x)$ はデータ数 N の DFT の結果なので仮定より式 (4.12) と式 (4.13) によって表すことができます。$f'(t) = f(2t)$ と置き換えたことを考えると $f'_{\frac{N}{2^k}}(x)$ は

$$
\begin{aligned}
& f'_{\frac{N}{2^k}}(2^k b_0 + \dots + b_k) = f'(C_2(\dots C_2(b_k, b_{k-1})\dots, b_0)) \\
\rightleftharpoons\ & f'_{\frac{N}{2^k}}(2^k b_0 + \dots + b_k) = f(2C_2(\dots C_2(b_k, b_{k-1})\dots, b_0))
\end{aligned}
$$

と書くことができます。同様に，$f''_{\frac{N}{2^k}}$ も

$$
f''_{\frac{N}{2^k}}(2^k b_0 + \dots + b_k) = f(2C_2(\dots C_2(b_k, b_{k-1})\dots, b_0) + 1)
$$

....

4.2　逆変換

逆変換の式は....