高速 Fourier 変換の概略メモ

大浦拓哉

1　FFT の基本的な考え方

高速 Fourier 変換アルゴリズム (FFT) が一般に知られるようになったのは，1965 年の J.W.Cooley と J.W.Tukey による短い論文 [3] からとされている．FFT があまり知られていなかったころは，N 点の離散 Fourier 変換 (DFT) を計算するためには N^2 回の計算が必要であると信じられてきた．しかし，FFT を用いると $N \log N$ に比例する計算で済む．この FFT の基本原理は，簡単な添字の変換で大きなサイズの DFT を計算が楽な小さな DFT に分解するという考えに基づく．

まず，N 点の DFT

$$A_k = \sum_{j=0}^{N-1} a_j W_N^{jk}, \quad W_N = \mathrm{e}^{-2\pi \mathrm{i}/N} \tag{1.1}$$

を素直に計算する場合を考える．この場合，A_0 から A_{N-1} までの各項の計算に N 回の乗算が入るため，全体で N^2 回の乗算が必要となる．しかし，もし N が 2 で割り切れるならば，添字 k を偶数と奇数に分けることで N 点の DFT は二つの $N/2$ 点の DFT

$$A_{2k} = \sum_{j=0}^{N/2-1} (a_j + a_{N/2+j}) W_{N/2}^{jk} \tag{1.2}$$

$$A_{2k+1} = \sum_{j=0}^{N/2-1} (a_j - a_{N/2+j}) W_N^j W_{N/2}^{jk} \tag{1.3}$$

に容易に分解できる．$N/2$ 点の DFT は素直に計算して $N^2/4$ 回の乗算で実行できるので，この分解で計算量は約半分に減ることになる．さらに，この分解を 2 回 3 回と繰り返せば計算量は約 1/4，1/8 と激減する．これが Cooley-Tukey FFT(正確には，基数 2, 周波数間引き Cooley-Tukey FFT) の基本的な考え方になる．

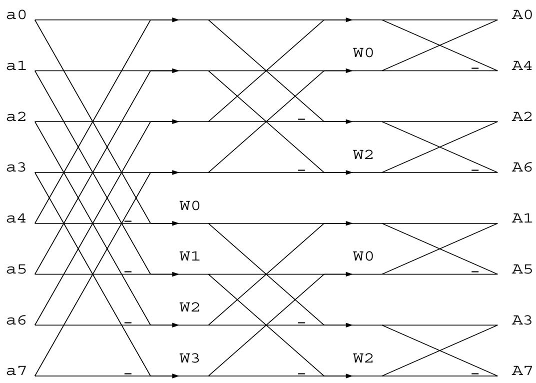

Fig. 1: 基数 2 周波数間引き FFT のデータフロー

この分解を $\log_2 N$ 行い，1 点の自明な DFT になるまで行ったときの計算量を考える．この分解自体には各々の段で W_N^j を乗ずる $N/2$ 回の複素数乗算と N 回の複素数加算が必要で，複素数乗算回数は $(N/2)\log_2 N$ に減少する．したがって，浮動小数点演算の量は $N \log_2 N$ のオーダーとな

る．これは，Cooley-Tukey FFT の典型的な演算量で，様々な FFT の演算量の削減アルゴリズムは，基本的にこのオーダーの比例定数と，$N\log_2 N$ より低い次数の項を小さくするものである．

2　FFT の分解の方法

ここでは FFT の添字によるより一般的な分解方法について調べる．まず，N が $N = N_1N_2$ と因数分解できると仮定する．このとき，(1.1) 式の添字 j を次の二つの添字 j_1，j_2 に $j_1 = 0, 1, 2, \ldots, N_1 - 1$, $j_2 = 0, 1, 2, \ldots, N_2 - 1$ に置き換える．次に，J_1, J_2 をある自然数として，j_1, j_2 から j に変換する写像

$$j \equiv (J_1j_1 + J_2j_2) \bmod N \tag{2.1}$$

を定義する．まず第一に，この写像は 1 対 1 とならなければならないが，このための必要十分条件は，p, q をある自然数とするとき，

1. N_1 と N_2 が互いに素の場合：
   $J_1 = pN_2$, $J_2 = qN_1$ の少なくとも一方が満たされかつ $\gcd(J_1, N_1) = \gcd(J_2, N_2) = 1$
2. N_1 と N_2 が互いに素ではない場合：
   $J_1 = pN_2$ かつ $J_2 \bmod N_1 \not\equiv 0$ かつ $\gcd(p, N_1) = \gcd(J_2, N_2) = 1$　または
   $J_1 \bmod N_2 \not\equiv 0$ かつ $J_2 = qN_1$ かつ $\gcd(J_1, N_1) = \gcd(q, N_2) = 1$

である [1]．さらに，添字 k に関しても同様の写像

$$k \equiv (K_1k_1 + K_2k_2) \bmod N \tag{2.2}$$

を定義し，(1.1) 式に対してこれらの変換を適用すると，次のようになる．

$$A_{K_1k_1+K_2k_2} = \sum_{j_2=0}^{N_2-1}\sum_{j_1=0}^{N_1-1} a_{J_1j_1+J_2j_2} W_N^{J_1K_1j_1k_1} W_N^{J_1K_2j_1k_2} W_N^{J_2K_1j_2k_1} W_N^{J_2K_2j_2k_2} \tag{2.3}$$

しかし，まだこのままでは演算ブロックの順序が入れ換えることができず，小さな DFT に分解することはできない．式の中の二番目と三番目の W の項が邪魔なのである．もし，条件

$$J_1K_2 \equiv 0 \bmod N \text{ または } J_2K_1 \equiv 0 \bmod N \text{ の少なくとも一方} \tag{2.4}$$

が成り立つのならば，(2.3) 式は N_1 と N_2 の二つの小さな DFT に分解されることがわかる．これらの条件を満たす例として，次の二種類の分解が考えられる．

1. N_1 と N_2 が互いに素の場合：
$$J_1 = N_2 \text{ かつ } J_2 = N_1 \text{ かつ } K_1 = N_2 \text{ かつ } K_2 = N_1$$
2. N_1 と N_2 が任意の場合：
$$J_1 = N_2 \text{ かつ } J_2 = 1 \text{ かつ } K_1 = 1 \text{ かつ } K_2 = N_1$$
   または
$$J_1 = 1 \text{ かつ } J_2 = N_1 \text{ かつ } K_1 = N_2 \text{ かつ } K_2 = 1$$

第一の場合は N_1 と N_2 が互いに素の場合のみに用いられる分解で，(2.3) 式の W の項を二つ消去して N_1 と N_2 の二次元 DFT に分解する．この分解は，N_1, N_2 を互いに素になるように選ぶ必要があるが，分解に必要な演算量はゼロである．しかし，分解しきれずに残った DFT はたいてい素数の長さであり，ある程度の計算量は必要になる．この分解による FFT は素因数 FFT [5, 2, 7] や Winograd DFT アルゴリズム [5, 9] に用いられる．一方，第二の分解は N_1 と N_2 は任意でよい代わりに (2.3) 式の W の項は一つしか消去されず，N_1 と N_2 の DFT への分解に W を乗算する演算 (回転因子の乗算) が必要になる．しかし，N_1 または N_2 を DFT が容易に計算できる数に固定できるため，分解以外に必要な計算量は少なくなる．この分解による FFT は Cooley-Tukey FFT [3] であり，N_1 を固定して分解を再帰的に繰り返すのが基本アルゴリズムである．このとき N_1 を基数といい，(2.3) 式の二項目の W を消し去るのが周波数間引き，三項目の W を消し去るのが時間間引きアルゴリズムと呼ぶ．この Cooley-Tukey FFT には多くの種類があり，通常の基数 2 の FFT，任意基数 FFT，混合基数 FFT，演算量が少ないとされる Split-Radix FFT [4, 6, 8] などがあげられる．

参考文献

[1] C. Burrus, *Index Mappings for Multidimensional Formulation of the DFT and Convolution*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.25 No.3 (1977), 239–242.

[2] C. Burrus, P. Eschenbacher, *An In-Place, In-Order Prime Factor FFT Algorithm*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.29 No.4 (1981), 806–817.

[3] J. Cooley, J. Tukey, *An Algorithm for the Machine Calculation of Complex Fourier Series*, Mathematics of Computation, Vol.19 (1965), 297–301.

[4] P. Duhamel, H. Hollmann, *Split-Radix FFT Algorithm*, Electronics Letters, Vol.20 (1984), 14–16.

[5] D. Kolba, *A Prime Factor FFT Algorithm Using High-Speed Convolution*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.25 No.4 (1977), 281–294.

[6] H. Sorensen, M. Heideman, C. Burrus, *On Computing the Split-Radix FFT*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.34 No.1 (1986), 152–156.

[7] C. Temperton, *Implementation of a Self-Sorting In-Place Prime Factor FFT Algorithm*, Journal of Computational Physics, vol.58 (1985), 283–299.

[8] M. Vetterli, P. Duhamel, *Split-Radix FFT Algorithms for Length-p^m DFT's*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.34 No.1 (1989), 57–64.

[9] S. Winograd, *On Computing the Discrete Fourier Transform*, Mathematics of Computation, Vol.32 (1978), 175–199.