

高速フーリエ変換、Fast Fourier Transform (FFT)

平成23年7月19日

古川 浩

FFT とは、離散フーリエ変換(DFT)もしくは逆離散フーリエ変換(IDFT)の計算に要する演算量を削減することを目的とする高速行列計算アルゴリズムである。

N は2のべき乗で与えられるものとする。 N 次の時系列信号を $f[n]$ とする。 $f[n]$ の DFT を $F[m]$ とおくと、 $F[m]$ は $f[n]$ より以下の式で与えられる。

$$F[m] = \sum_{n=0}^{N-1} f[n] W_N^{mn} \quad \dots (1)$$

ここで、 $W_N = e^{-2\pi j/N}$ で与えられる。(1)式は以下のように分解できる。

$$\begin{aligned} F[m] &= \sum_{n=0}^{N/2-1} f[2n] W_N^{m2n} + \sum_{n=0}^{N/2-1} f[2n+1] W_N^{m2(n+1)} \\ &= \sum_{n=0}^{N/2-1} f[2n] W_N^{m2n} + W_N^m \sum_{n=0}^{N/2-1} f[2n+1] W_N^{m2n} \end{aligned}$$

ここで、 $W_N^{m2n} = (W_N^2)^{mn} = W_{N/2}^{mn}$ より、

$$F[m] = \underbrace{\sum_{n=0}^{N/2-1} f[2n] W_{N/2}^{mn}} + W_N^m \underbrace{\sum_{n=0}^{N/2-1} f[2n+1] W_{N/2}^{mn}} \quad \dots (2)$$

が導かれる。(2)式の一重下線部は $f[n]$ の偶数項の DFT、また二重下線部は奇数項の DFT に他ならない。つまり、DFT は被 DFT 系列の偶数項の DFT と奇数項の DFT の重みづけ和で与えられることが分る。

(1)式で与えられる DFT を図1のような模式図で表してみよう。ここで、 $N=8$ とする。

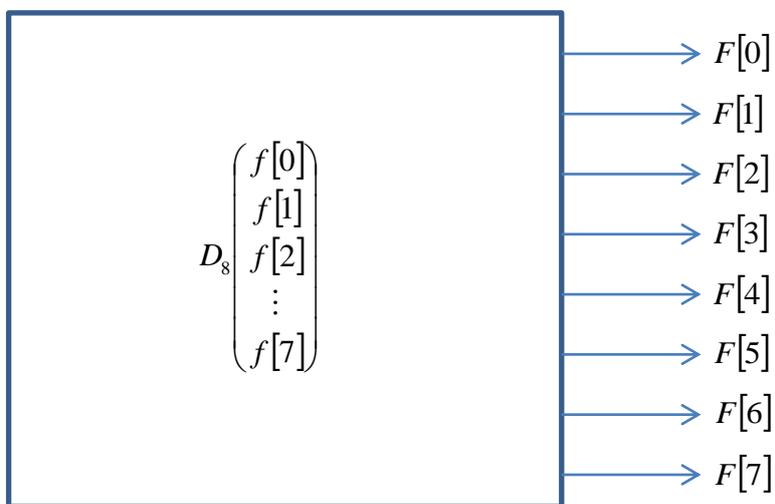


図1 DFT を模式的に表した図

D_8 は 8 次の DFT 行列を表しており、 N 次の DFT 行列は以下で定義される。

$$D_N = \begin{pmatrix} W_N^{0 \cdot 0} & W_N^{0 \cdot 1} & \dots & W_N^{0 \cdot (N-1)} \\ W_N^{1 \cdot 0} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ W_N^{(N-1) \cdot 0} & W_N^{(N-1) \cdot 1} & \dots & W_N^{(N-1) \cdot (N-1)} \end{pmatrix}$$

図1の四角で囲った部分は8次の正方行列に8行の列ベクトルを掛け合わせる演算処理を表している。この場合、複素乗算処理が $8 \times 8 = 64$ 回必要となる。

一方、(2)式によるDFTを図1と同様の表現法に従い表現すると図2のようになる。

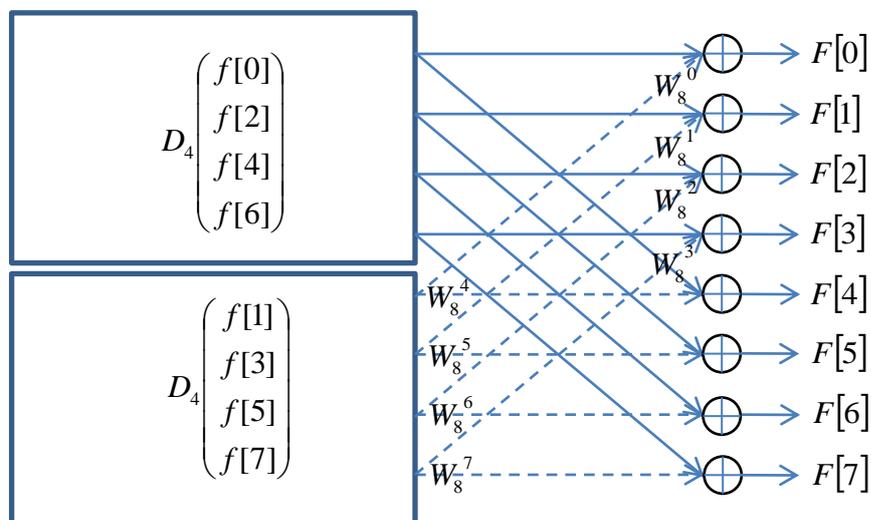


図2 DFT 演算を偶数項と奇数項それぞれの DFT の重みづけ和で表した場合の処理模式図

図2における複素乗算回数を考えてみよう。図2中の四角で囲った2つの行列演算における複素乗算回数は、それぞれ $4 \times 4 = 16$ 回となる。これらの行列演算結果を重み付け加算する部分の複素乗算回数は8回である。すなわち、合計の複素乗算回数は、

$$4 \times 4 + 4 \times 4 + 8 = 40 \text{ 回}$$

となる。図1の場合は64回だった複素乗算回数が、偶数項と奇数項のDFTに分解し、これらの結果を重み付け加算することで、40回へと削減されるのである。

図2中の四角で囲った2つの行列演算部分を、それぞれ、さらに偶数項と奇数項のDFTへと分解してみよう。図3に結果を示す。

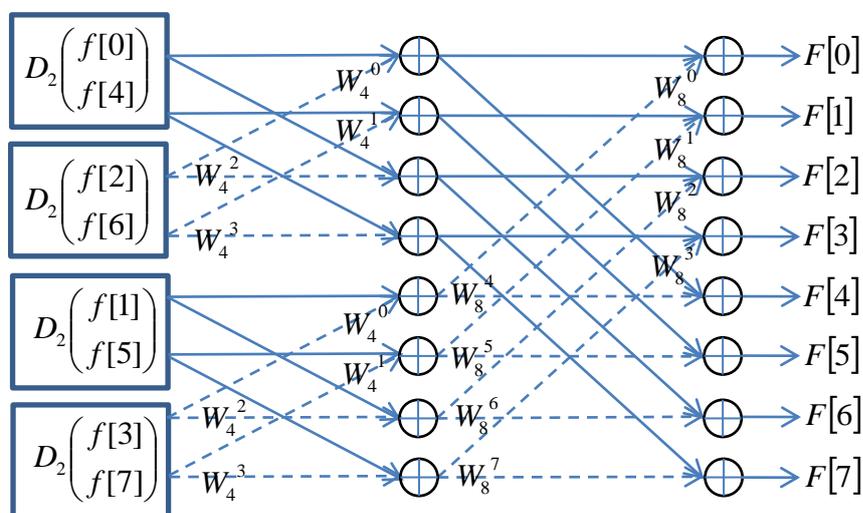


図3 図2の DFT ブロックをさらにそれぞれ偶数項と奇数項の DFT の重みづけ和で表した場合の処理模式図

必要な複素乗算回数を評価する。図3より、4つの DFT ブロックはそれぞれ $2 \times 2 = 4$ 回の複素乗算が必要である。4つの DFT の後に続く2段の重み付け加算処理部分の複素乗算回数は、それぞれ8回である。すなわち、合計の複素乗算回数は、

$$2 \times 2 \times 4 + 8 + 8 = 32 \text{ 回}$$

となる。

同様に図3の4つの DFT ブロックをさらに偶数項と奇数項に分けて重み付け加算する。図4に結果を示す。

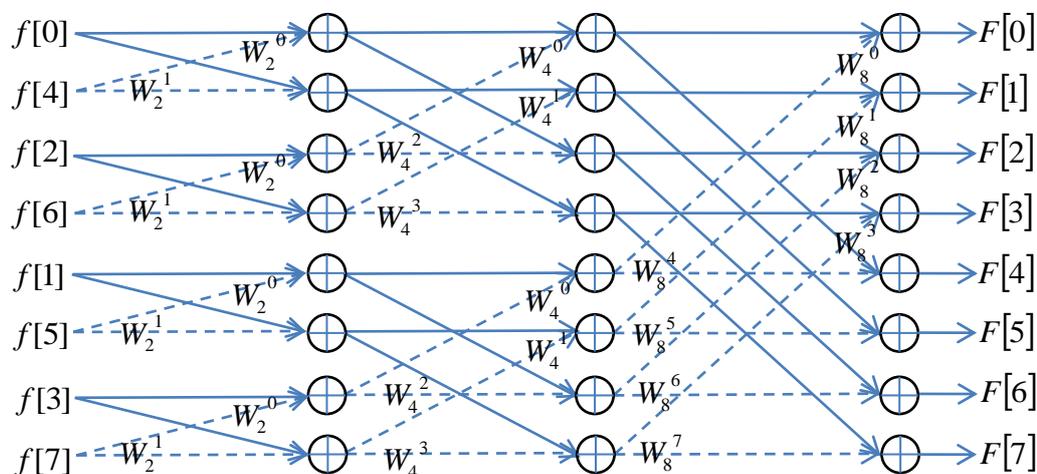


図4 図3の DFT ブロックをさらにそれぞれ偶数項と奇数項の DFT の重みづけ和で表した場合の処理模式図

図4より、この場合の乗算回数を計算すると、

$$8 + 8 + 8 = 24 \text{ 回}$$

となる。図1で必要な複素乗算回数は64回であったのと比較し、図4では同じ演算を行うにも関わらず複素乗算回数は24回にまで減らすことが可能となった。

上記の議論を任意の N (ただし、 N は2のべき乗で与えられるものとする) へと拡張する。FFT を用いない場合の DFT 演算に必要な複素乗算回数は、

$$N^2$$

であるのに対して、FFT を用いる場合に必要な複素乗算回数は次式で与えることができる。

$$N \log_2 N$$

疎行列分解の視点で眺めた FFT アルゴリズム

別の視点で上述の FFT を眺めてみよう。図 1 に示した DFT 演算は次の行列演算を表している。

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \\ F[4] \\ F[5] \\ F[6] \\ F[7] \end{pmatrix} = D_8 \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ f[3] \\ f[4] \\ f[5] \\ f[6] \\ f[7] \end{pmatrix} = \begin{pmatrix} W_8^{0\cdot0} & W_8^{0\cdot1} & W_8^{0\cdot2} & W_8^{0\cdot3} & W_8^{0\cdot4} & W_8^{0\cdot5} & W_8^{0\cdot6} & W_8^{0\cdot7} \\ W_8^{1\cdot0} & W_8^{1\cdot1} & W_8^{1\cdot2} & W_8^{1\cdot3} & W_8^{1\cdot4} & W_8^{1\cdot5} & W_8^{1\cdot6} & W_8^{1\cdot7} \\ W_8^{2\cdot0} & W_8^{2\cdot1} & W_8^{2\cdot2} & W_8^{2\cdot3} & W_8^{2\cdot4} & W_8^{2\cdot5} & W_8^{2\cdot6} & W_8^{2\cdot7} \\ W_8^{3\cdot0} & W_8^{3\cdot1} & W_8^{3\cdot2} & W_8^{3\cdot3} & W_8^{3\cdot4} & W_8^{3\cdot5} & W_8^{3\cdot6} & W_8^{3\cdot7} \\ W_8^{4\cdot0} & W_8^{4\cdot1} & W_8^{4\cdot2} & W_8^{4\cdot3} & W_8^{4\cdot4} & W_8^{4\cdot5} & W_8^{4\cdot6} & W_8^{4\cdot7} \\ W_8^{5\cdot0} & W_8^{5\cdot1} & W_8^{5\cdot2} & W_8^{5\cdot3} & W_8^{5\cdot4} & W_8^{5\cdot5} & W_8^{5\cdot6} & W_8^{5\cdot7} \\ W_8^{6\cdot0} & W_8^{6\cdot1} & W_8^{6\cdot2} & W_8^{6\cdot3} & W_8^{6\cdot4} & W_8^{6\cdot5} & W_8^{6\cdot6} & W_8^{6\cdot7} \\ W_8^{7\cdot0} & W_8^{7\cdot1} & W_8^{7\cdot2} & W_8^{7\cdot3} & W_8^{7\cdot4} & W_8^{7\cdot5} & W_8^{7\cdot6} & W_8^{7\cdot7} \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ f[3] \\ f[4] \\ f[5] \\ f[6] \\ f[7] \end{pmatrix} \dots (3)$$

これに対して、図 2 に示した模式図は上式の行列演算中の DFT 行列が次に示す 2 つの正
 方形列へと分解しうることを表している。

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \\ F[4] \\ F[5] \\ F[6] \\ F[7] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & W_8^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^3 \\ 1 & 0 & 0 & 0 & W_8^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^5 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^6 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^7 \end{pmatrix} \begin{pmatrix} D_4 & 0 \\ 0 & D_4 \end{pmatrix} \begin{pmatrix} f[0] \\ f[2] \\ f[4] \\ f[6] \\ f[1] \\ f[3] \\ f[5] \\ f[7] \end{pmatrix} \dots (4)$$

さらに図 3 は、(4)式中の正方形列 $\begin{pmatrix} D_4 & 0 \\ 0 & D_4 \end{pmatrix}$ がさらに 2 つの行列へと分解できることを表している。

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \\ F[4] \\ F[5] \\ F[6] \\ F[7] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & W_8^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^3 \\ 1 & 0 & 0 & 0 & W_8^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^5 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^6 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^7 \end{pmatrix} \begin{pmatrix} 1 & 0 & W_4^0 & 0 & & & & \\ 0 & 1 & 0 & W_4^1 & & & & \\ & & & & 0 & & & \\ 1 & 0 & W_4^2 & 0 & & & & \\ 0 & 1 & 0 & W_4^3 & & & & \\ & & & & & 1 & 0 & W_4^0 & 0 \\ & & & & & & 0 & 1 & 0 & W_4^1 \\ & & & & & & & & 1 & 0 & W_4^2 & 0 \\ & & & & & & & & & 0 & 1 & 0 & W_4^3 \end{pmatrix}$$

$$\begin{pmatrix} D_2 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ 0 & 0 & D_2 & 0 \\ 0 & 0 & 0 & D_2 \end{pmatrix} \begin{pmatrix} f[0] \\ f[4] \\ f[2] \\ f[6] \\ f[1] \\ f[5] \\ f[3] \\ f[7] \end{pmatrix}$$

... (5)

(5)式より、最終的には以下の式が導かれる。

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ F[3] \\ F[4] \\ F[5] \\ F[6] \\ F[7] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & W_8^0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^3 \\ 1 & 0 & 0 & 0 & W_8^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & W_8^5 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^6 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & W_8^7 \end{pmatrix} \begin{pmatrix} 1 & 0 & W_4^0 & 0 & & & & \\ 0 & 1 & 0 & W_4^1 & & & & \\ & & & & 0 & & & \\ 1 & 0 & W_4^2 & 0 & & & & \\ 0 & 1 & 0 & W_4^3 & & & & \\ & & & & & 1 & 0 & W_4^0 & 0 \\ & & & & & & 0 & 1 & 0 & W_4^1 \\ & & & & & & & & 1 & 0 & W_4^2 & 0 \\ & & & & & & & & & & 0 & 1 & 0 & W_4^3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & W_2^0 & & & & & & \\ 1 & W_2^1 & & & & & & \\ & & 1 & W_2^0 & & & & \\ & & & & 1 & W_2^1 & & \\ & & & & & & 1 & W_2^0 \\ & & & & & & & & 1 & W_2^1 \\ & & & & & & & & & & 1 & W_2^0 \\ & & & & & & & & & & & & 1 & W_2^1 \end{pmatrix} \begin{pmatrix} f[0] \\ f[4] \\ f[2] \\ f[6] \\ f[1] \\ f[5] \\ f[3] \\ f[7] \end{pmatrix}$$

... (6)

(6)式の行列演算を模式的に表現したものが図4に他ならない。(6)式が示す通り、当初のDFT行列は3つの正方行列へと分解された。これら3つの正方行列は、それぞれ多くの0成分を含むことから疎行列と呼ばれる。このように、ある行列を複数の疎行列に分解することを疎行列分解と呼ぶ。疎行列は0成分を多く含むことから、同行列を含む行列演算では乗算回数を大きく減らすことができる。つまりFFTとは、DFT行列を疎行列分解することによって演算量を削減する手法なのである。

(3)式の行列演算における複素乗算回数は8 x 8 = 64回であるのに対して、(6)式では、8 + 8 + 8 = 24回へと減らせることが直ちに理解される。

なお、(6)式における被DFT列ベクトルの並びを降順に並び替えるには、以下の正方行列を挿入すればよい。

