

高速フーリエ変換の基礎

東京大学 毛利研究室 修士一年 佐久田昌博

平成 19 年 7 月 13 日

この文書のコピーは <http://mohri1.t.u-tokyo.ac.jp/japanese/sakuta/fft.pdf> にあります。

1 フーリエ級数展開

フーリエ級数展開

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos \frac{2\pi}{T} kt + b_k \sin \frac{2\pi}{T} kt \right) \quad (1)$$

$$a_k = \frac{2}{T} \int_{-\pi}^{\pi} f(t) \cos \frac{2\pi}{T} kt \, dt \quad (k = 0, 1, 2, \dots) \quad (2)$$

$$b_k = \frac{2}{T} \int_{-\pi}^{\pi} f(t) \sin \frac{2\pi}{T} kt \, dt \quad (k = 1, 2, \dots) \quad (3)$$

複素フーリエ級数展開

$$f(t) = \sum_{k=-\infty}^{\infty} C_k e^{j\omega_0 kt} \quad (4)$$

$$C_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j\omega_0 kt} \, dt \quad (\omega_0 = 2\pi/T) \quad (5)$$

2 フーリエ変換

複素フーリエ級数展開の区間を無限大に発散させることによって、フーリエ変換が得られる。ここで $F(\omega)$ が一般には複素数になることに注意されたい。

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (6)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (7)$$

尚、係数の $1/2\pi$ はここに示した通りの定義と違って、下段の式についている場合が参考文献によってはありうるが、どちらが正しいということはなく、使いやすい方を選べばよい。

3 離散フーリエ変換 (DFT)

複素フーリエ級数展開を連続関数をサンプリングした有限の要素からなる空間同士の変換へと書き換えると、離散フーリエ変換が得られる。これは要素の細かさを極限にまで細かくすればフーリエ変換と同等である。

$$f_i = \sum_{k=0}^{N-1} C_k e^{j(2\pi/N)ki} \quad (i = 0, 1, 2, \dots, N-1) \quad (8)$$

$$C_k = \frac{1}{N} \sum_{i=0}^{N-1} f_i e^{-j(2\pi/N)ki} \quad (k = 0, 1, 2, \dots, N-1) \quad (9)$$

4 高速フーリエ変換 (FFT)

高速フーリエ変換は、離散フーリエ変換を高速に行うアルゴリズムの一種である。DFT では指数関数と f_i の積が $O(N^2)$ で現れてくるため、 N が大きくなると非常に計算に時間がかかるようになる。これに対し、以下に述べる FFT では積を取る回数が $O(N \log N)$ になる。これは N が大きい時には遥かに DFT よりも計算回数が少なくて済むということである。

FFT には他に、サンプル数が 2 のべき乗でなければならないという特徴がある。

以下では、簡単な具体例から FFT がどのようなアルゴリズムであるかを述べる。

例えば、4 サンプルの DFT において、 $e^{-j(2\pi/N)i} = w^i$ において、 C_k への変換を行列で表すと

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 & w^0 & w^0 \\ w^0 & w^1 & w^2 & w^3 \\ w^0 & w^2 & w^4 & w^6 \\ w^0 & w^3 & w^6 & w^9 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad (10)$$

となる。ここで積を偶数と奇数に分けてみると

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^4 \\ w^0 & w^6 \end{pmatrix} \begin{pmatrix} f_0 \\ f_2 \end{pmatrix} + \begin{pmatrix} w^0 & w^0 \\ w^1 & w^3 \\ w^2 & w^6 \\ w^3 & w^9 \end{pmatrix} \begin{pmatrix} f_1 \\ f_3 \end{pmatrix} \quad (11)$$

と書けることが分かる。

ここで右辺の第 2 項の左に掛かっている 2×4 行列が

$$\begin{pmatrix} w^0 & w^0 \\ w^1 & w^3 \\ w^2 & w^6 \\ w^3 & w^9 \end{pmatrix} = \begin{pmatrix} w^0 w^0 & w^0 w^0 \\ w^1 w^0 & w^1 w^2 \\ w^2 w^0 & w^2 w^4 \\ w^3 w^0 & w^3 w^6 \end{pmatrix} = \begin{pmatrix} w^0 & 0 & 0 & 0 \\ 0 & w^1 & 0 & 0 \\ 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & w^3 \end{pmatrix} \begin{pmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^4 \\ w^0 & w^6 \end{pmatrix} \quad (12)$$

と変形できることに注目すると

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^4 \\ w^0 & w^6 \end{pmatrix} \begin{pmatrix} f_0 \\ f_2 \end{pmatrix} + \begin{pmatrix} w^0 & 0 & 0 & 0 \\ 0 & w^1 & 0 & 0 \\ 0 & 0 & w^2 & 0 \\ 0 & 0 & 0 & w^3 \end{pmatrix} \begin{pmatrix} w^0 & w^0 \\ w^0 & w^2 \\ w^0 & w^4 \\ w^0 & w^6 \end{pmatrix} \begin{pmatrix} f_1 \\ f_3 \end{pmatrix} \quad (13)$$

$$= \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_2 \end{pmatrix} + \begin{pmatrix} w^0 & 0 & 0 & 0 \\ 0 & w^1 & 0 & 0 \\ 0 & 0 & -w^0 & 0 \\ 0 & 0 & 0 & -w^1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} f_1 \\ f_3 \end{pmatrix} \quad (14)$$

となる。ここで、太字にした部分に全く同じ掛け算が出てきているのがわかる。このような類似性に着目すると

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} \alpha_0 + \alpha_1 \\ \alpha_2 + \alpha_3 \\ \alpha_0 - \alpha_1 \\ \alpha_2 - \alpha_3 \end{pmatrix} \quad \begin{cases} \alpha_0 = f_0 + w^0 f_2 \\ \alpha_1 = w^0(f_1 + w^0 f_3) \\ \alpha_2 = f_0 - w^0 f_2 \\ \alpha_3 = w^1(f_1 - w^0 f_3) \end{cases} \quad (15)$$

とまとめられる。

これを図で表すと図 1 のようになる。点線は -1 倍して足すことを示している。この計算は、その形からバタフライ演算と呼ばれている。

素直に DFT を行った場合、4 サンプルでは 16 回の乗算が必要になるが、この FFT では 4 回で済んでいる。

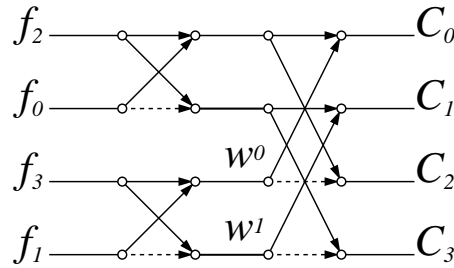


図 1: バタフライ演算による 4 サンプル FFT

式の量が膨大になるため具体的には示さないが、8 サンプルの FFT においても同様に偶数と奇数のグループに分けると、二つの 4 サンプル FFT に帰着できる。さらに 16 サンプ

ルの場合は 8 サンプルが二つ、というふうに、一段低い FFT によって次段の FFT が行われる。このため、FFT はプログラム手続きの再起呼び出しを使って実装されやすい。

ちなみに、8 サンプル FFT についての図だけを図 2 に示す。ここで $w^i = e^{-j(2\pi/8)i}$ である。

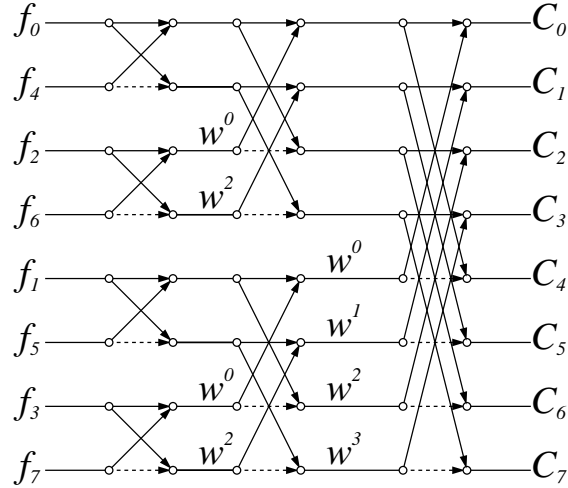


図 2: バタフライ演算による 8 サンプル FFT

5 フーリエ変換の応用

5.1 パワースペクトル

フーリエ変換の結果 $F(\omega)$ の絶対値の自乗をとったものをパワースペクトルという。これは自身の共役複素数との積として定義されることもあるが、意味は同じである。

$$P(\omega) = |F(\omega)|^2 = F^*(\omega) F(\omega) \quad (16)$$

電圧・電流や音圧などの信号をそのエネルギーに比例するスケールにとったものと言える。

尚、これは変換元の関数の自己相関関数 (auto-correlation) をフーリエ変換したものと同じ関数である。自己相関関数は相互相関関数 (cross-correlation) の定義

$$R_{fg}(\tau) = \langle f(t), g(t + \tau) \rangle = \frac{1}{T} \int_0^T f(t)g(t + \tau)dt \quad (17)$$

において、 $R_{ff} = \langle f(t), f(t + \tau) \rangle$ のことである。

5.2 クロススペクトル

二つの関数のフーリエ変換 $F(\omega)$ 、 $G(\omega)$ について、クロススペクトルは次のように定義される。

$$P_c(\omega) = F^*(\omega) G(\omega) \quad (18)$$

尚、これは変換元の関数の相互相関関数をフーリエ変換したものと同一関数である。

$$\mathcal{F}[R_{fg}] = \mathcal{F}[\langle f(t), g(t + \tau) \rangle] = F^*(\omega) G(\omega) \quad (19)$$

5.3 位相スペクトル

フーリエ変換の複素平面上の位相を周波数の関数としてとったものを位相スペクトルという。

$$\Phi(\omega) = \arg(F(\omega)) = \tan^{-1} \frac{\Re[F]}{\Im[F]} \quad (20)$$

5.4 畳み込み関数

畳み込み関数、畳み込み積分、あるいはコンボリューションと呼ばれる演算は、二つの任意の関数 $h(t)$ 、 $x(t)$ によって次のように定義されている。

$$y(t) = h * x = \int_0^\infty h(\tau) \cdot x(t - \tau) d\tau \quad (21)$$

ここで、 $h(t)$ と $x(t)$ のフーリエ変換 $H(\omega)$ 、 $X(\omega)$ を使うと、畳み込み関数は単なる積で表せる。

$$y(t) = h * x = H \cdot X \quad (22)$$

証明は略。

6 線形システム解析

線形システムとは、入力と出力の関係が線形の微分方程式で表されるような系のことである。実際の現象には非線形なシステムも数多いが、線形とみなして解析しうるシステムも数多い。このような系ではフーリエ変換を応用した解析が有効である。

入力 $x(t)$ に対して出力 $y(t)$ が得られる線形システム h を考える。このとき、出力はある関数 $h(t)$ による畳み込み関数 $y(t) = h * x$ で表せる。ここに入力としてデルタ関数を入れると、(21) から

$$y(t) = h * \delta = \int_0^{\infty} h(\tau) \cdot \delta(t - \tau) d\tau = h(t) \quad (23)$$

となり、出力は関数 $h(t)$ そのものになる。デルタ関数に対する出力であることから、 $h(t)$ をインパルス応答という（グリーン関数ともいう）。現実にはデルタ関数のように時間が無限小で強さが無限大のような信号は作り出せないので、ピークの高い短時間パルスで近似することになる。

このことから分かるのは、インパルス応答がわかれば、一般の入力 $x(t)$ に対する出力が畳み込み積分によって予測できるということである。つまり、インパルス応答が系そのものを表しているといっている。

ここまでの議論は実空間の時間領域に関するものだったが、これをフーリエ変換した周波数領域で考えると、解析に有用な結果が得られる。まず、入力 $x(t)$ をフーリエ変換した結果 $X(\omega)$ と出力 $y(t)$ をフーリエ変換した結果 $Y(\omega)$ の間には、インパルス応答 $h(t)$ をフーリエ変換した結果 $H(\omega)$ を掛けただけという関係がある。

$$Y(t) = H \cdot X \quad (24)$$

これは (22) から導かれる結果である。このことから逆に、入力と出力のフーリエ変換からインパルス応答を求めることができる。

$$h = \mathcal{F}^{-1}[H] = \mathcal{F}^{-1}[Y/X] \quad (25)$$

また、 $X(\omega)$ のパワースペクトル $P_X(\omega) = X(\omega) X^*(\omega)$ を入力として系に通すと、

$$H \cdot P_X = H \cdot X \cdot X^* = Y \cdot X^* = P_{XY} \quad (26)$$

となり、クロススペクトルが求まる。このことから、時間領域では入力 $x(t)$ の自己相関関数 $c_i(\tau)$ を入力としたときの出力 $h * c_i$ が相互相関関数 $c_{io}(\tau) = \langle x(t), y(t + \tau) \rangle$ になっていることが類推できる。

また、クロススペクトル $P_{XY}(\omega)$ を更に系に通してやると、

$$H \cdot P_{XY} = H \cdot Y \cdot X^* = Y \cdot Y^* = P_Y \quad (27)$$

となり、出力 $Y(\omega)$ のパワースペクトルが得られる。このことから、時間領域では入力と出力の相互相関関数 $c_{io}(\tau)$ をまたシステムに入力した出力 $h * c_{io}$ が出力の自己相関関数 $c_o(\tau) = \langle y(t), y(t + \tau) \rangle$ になっていることが類推できる。