

フーリエ変換サブルーチン・パッケージの作成

名大工学部情報 鳥 店 達 生

1. はじめに

周知のように大量のデータの離散型フーリエ変換において高速フーリエ変換 (FFT) は欠かせない手法になっている。

入力データが実数で、かつ対称 (歪対称) ならば、離散型 cosine (sine) 変換となる。入力データが複素数の FFT において、入力データの实数性、対称性を利用するならば、演算回数および作業領域を $1/4$ に減らすことができる。

高速 cosine (sine) 変換は、関数の N エビニエフ級数展開にも用いられる。関数のフーリエ級数展開の場合、入力は一変数の関数と許される誤差限界 $\epsilon > 0$ であり、出力は展開項数 $N(\epsilon)$ と $N(\epsilon)$ 個の離散型フーリエ係数である。

筆者は、以前 cosine (sine) 変換の FFT のプログラムを作成したが、そこでは入力データと同じ大まな作業

領域を使用していた¹⁾。今回、入力データをビット逆転²⁾によって並べ換え、補助的な作業領域を用いない高速 cosine (sine) 変換のプログラムをつくった。²⁾ 入力データの実数性、対称性を利用してつつ重ね書き (over write) しているのが、算法は若干複雑となる。

以下にパッケージの要素となっているプログラム名を記す。

- (1) 中点公式に基づく高速 cosine 変換 (基底 2)
 - (2) 中点公式に基づく高速 sine 変換 (基底 2)
 - (3) 三角関数表 (ビット逆転の順序)
 - (4) 台形公式に基づく高速 cosine 変換 (基底 2)
 - (5) 台形公式に基づく高速 sine 変換 (基底 2)
 - (6) 閉区間 $[0, \pi]$ で与えられた関数の cosine 級数展開
 - (7) 開区間 $(0, \pi)$ で与えられた関数の cosine 級数展開
 - (8) 開区間 $(0, \pi)$ で与えられた関数の sine 級数展開
 - (9) ギ 1 種 $4E$ ビ $3E$ フ多項式系による展開 (閉いた公式)
 - (10) ギ 2 種 $4E$ ビ $3E$ フ多項式系による展開 (開いた公式)
 - (11) ギ 3 種 $4E$ ビ $3E$ フ多項式系による展開 (閉いた公式)
 - (12) ギ 1 種 $4E$ ビ $3E$ フ多項式系による展開 (開いた公式)
 - (13) Clenshaw-Curtis 法による自動積分 (閉いた公式)
 - (14) Clenshaw-Curtis 法による自動積分 (開いた公式)
- これらの外に、各種の級数の値を求めたり、項別の微分 (

導関数), 不定積分など計算するプログラムを作成したが省略する.

これら一連のプログラムの基本となるのは, 中点公式に基づく高速 cosine (sine) 変換であるから, 次節で, これに関して説明する.

2. 中点公式に基づく高速 cosine 変換

周期 2π の複素数値関数 $X(t)$ に対する N 項の中点公式によるフーリエ変換とは $N=2^n$

$$B_k = \sum_{0 \leq j < 2^n} X\left(\frac{2\pi}{2^n}\left(j + \frac{1}{2}\right)\right) \exp\left(-\frac{2\pi i}{2^n} k \left(j + \frac{1}{2}\right)\right) \quad (1)$$

$$k = 0, 1, \dots, 2^n - 1$$

である. これに対する重ね書きの FFT の算法を示す.

準備として n ビットの正の整数

$$j = j_1 2^0 + j_2 2^1 + \dots + j_n 2^{n-1}$$

に対して $1 \leq j \leq 2^n$ の n ビットの数列

$$j^* = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$$

を定義する. 数列 $\{j^*\}$ は, $(0, 1)$ 上に一様分布する

Van der Corput 列の名で知られている. この数列は

$$(2j)^* = j^* / 2$$

$$(2j+1)^* = \frac{1}{2} + j^* / 2$$

の漸化式から生成される. j^* は j の n ビット 逆転に 2^{-n} をかけたものである. これを用いてフーリエ変換(1)は

$$B_k = \sum_{0 \leq j < 2^n} X(2\pi(j+2^n)^*) \exp(-2\pi i k(j+2^n)^*) \quad (2)$$

と表す変えられる. これは(1)と和の順序が異なるだけである.

ここで原素を $(2^l j)^*$ だけずらしながら, 2^l 項のフーリエ変換を

$$X^l(k+2^l j) \equiv \sum_{0 \leq p < 2^l} X(2\pi(p+2^l j+2^n)^*) \exp(-2\pi i k(p+2^l j+2^n)^*)$$

$$0 \leq k < 2^l, \quad 0 \leq j < 2^{n-l}, \quad 0 \leq l \leq n \quad (3)$$

とおく.

明らかに

$$X^0(j) = X(2\pi(j+2^n)^*) \quad , \quad 0 \leq j < 2^n$$

$$X^n(k) = B_k \quad , \quad 0 \leq k < 2^n$$

であるから, $X^0(j)$ を初期値として $X^n(k)$ を求める算法を導けばよい.

中点公式に基づくFFT …… 入力は複素数データ

$$X^l(k+2^l j) = X^{l-1}(k+2^l j) + X^{l-1}(k+2^{l-1}+2^l j)$$

$$X^l(k+2^{l-1}+2^l j) = \exp(-\pi i (j+2^{n-l})^*) \quad (4)$$

$$\cdot \{ X^{l-1}(k+2^l j) - X^{l-1}(k+2^{l-1}+2^l j) \}$$

$$0 \leq j < 2^{n-l}, \quad 0 \leq k < 2^{l-1}, \quad l = 0, 1, \dots, n.$$

求めるフーリエ係数 $B_k = X^n(k)$ のために要する複素数乗算回数は $\frac{1}{2} N \log_2 N$, $N=2^n$ である。

そこでテーブル(3)をFFTテーブルとよぶことにする。

(3)から(4)の導出は、少し面倒な計算である。

次に入力データが実数の場合の算法について述べる。入力データの实数性はFFTテーブルに、次の性質をもたす。

定理1. 入力データが実数ならば

$$X^l(2^l - k + 2^l j) = \bar{X}^l(k + 2^l j) \exp(-2\pi i (j+2^{n-l})^*)$$

$$0 < k < 2^l, \quad 0 \leq j < 2^{n-l}$$

$$X^l(2^l j) \text{ は実数}$$

が成り立つ。 $\bar{X}^l(\cdot)$ は $X^l(\cdot)$ の共役複素数。

証明 FFTテーブルの定義によつて

$$X^l(2^l - k + 2^l j) = \sum_{0 \leq p < 2^l} X(2\pi(p+2^l j+2^n)^*) \exp(-2\pi i$$

$$(2^{\ell} - k)(p + 2^{\ell}j + 2^n)^*$$

入力タームは実数であるから

$$= \sum_p \bar{X}(2\pi(p + 2^{\ell}j + 2^n)^*) \exp(2\pi i k(p + 2^{\ell}j + 2^n)^*) \\ \cdot \exp(-2\pi i 2^{\ell}(2^{\ell}j + 2^n)^*)$$

$$= \bar{X}^{\ell}(k + 2^{\ell}j) \exp(-\pi i (j + 2^{n-\ell})^*) \quad (\text{証明終})$$

FFTラベルにおいて, この性質を利用すれば, 添字 k の動く範囲を半分=減らすことができる.

入力タームが実数の場合のFFT

$$X^{\ell}(k + 2^{\ell}j) = X^{\ell-1}(k + 2^{\ell}j) + X^{\ell-1}(k + 2^{\ell-1} + 2^{\ell}j)$$

$$X^{\ell}(\cancel{2^{\ell-1}} - k + 2^{\ell}j) = W \{ \bar{X}^{\ell-1}(k + 2^{\ell}j) - \bar{X}^{\ell-1}(k + 2^{\ell-1} + 2^{\ell}j) \}$$

$$W \equiv \exp(-\pi i (j + 2^{n-\ell})^*) \quad (5)$$

$$0 < k < 2^{\ell-2}, \quad 0 \leq j < 2^{n-\ell}$$

$k=0$, $2^{\ell-2}$ の場合は特別扱いして以下の式を用いる.

$$X^{\ell}(2^{\ell}j) = X^{\ell-1}(2^{\ell}j) + X^{\ell-1}(2^{\ell-1} + 2^{\ell}j)$$

$$\exp(\pi i (j + 2^{n-\ell})) X^{\ell}(2^{\ell-1} + 2^{\ell}j) = X^{\ell-1}(2^{\ell}j) - X^{\ell-1}(2^{\ell-1} + 2^{\ell}j)$$

$$X^{\ell}(2^{\ell-2} + 2^{\ell}j) = \exp(-\frac{\pi}{2} (j + 2^{n-\ell})^*)$$

$$\cdot \{ X^{\ell-1}(2^{\ell-2} + 2^{\ell}j) \exp(\pi i (j + 2^{n-\ell})^*/2) - X^{\ell-1}(2^{\ell-2} + 2^{\ell-1} + 2^{\ell}j) \\ \cdot \exp(\pi i (1 + 2j + 2^{n-\ell+1})^*) \}$$

以上の算法によって重畳表まがでまることを示すために、
複素数 $X^l(k+2^l j)$, $k, j \geq 0$ の実数部, 虚数部に
 $k+2^l j$, $2^{l-1}-k+2^l j$ の番地を割当て

$$X^l(k+2^l j) \equiv X^l \begin{pmatrix} k+2^l j \\ 2^{l-1}-k+2^l j \end{pmatrix} \equiv X^l \begin{pmatrix} k & 0 & j \\ \bar{k} & 1 & j \end{pmatrix}$$

ただし $\bar{k} = 2^{l-1} - k$
と書く。 $k = 0$, 2^{l-1} を実数部, 虚数部に対応させる。
つまり, FFT (5) は次のように略記できる。

$$X^l \begin{pmatrix} k & 0 & j \\ \bar{k} & 1 & j \end{pmatrix} = X^{l-1} \begin{pmatrix} k & 0 & j \\ \bar{k} & 0 & j \end{pmatrix} + X^{l-1} \begin{pmatrix} k & 1 & j \\ \bar{k} & 1 & j \end{pmatrix} \quad (6)$$

$$X^l \begin{pmatrix} \bar{k} & 0 & j \\ k & 1 & j \end{pmatrix} = \bar{w} \left\{ \bar{X}^{l-1} \begin{pmatrix} k & 0 & j \\ \bar{k} & 0 & j \end{pmatrix} - \bar{X}^{l-1} \begin{pmatrix} k & 1 & j \\ \bar{k} & 1 & j \end{pmatrix} \right\}$$

ここで 0 または 1 は l ビット目にあたる。

k と j は, それぞれ $l-1$ ビット, $n-l$ ビットの正の整数をすべて動く。

関数 $X(t)$ が $X(t) = \bar{X}(-t)$ を満たすとき,
Hermitian 対称という。

定理 入力データが Hermitian 対称

$$X(2\pi(j+2^n)^*) = \bar{X}(2\pi(j+2^n)^*) \quad , \quad \bar{j} = 2^n - 1 - j$$

ならば, FFT ラーブルは対称

$$X^l(k+2^l \bar{j}) = \bar{X}^l(k+2^l j), \quad j = 2^{n-l} - 1 - \bar{j}$$

である。

系 入力 $\bar{x}$ が Hermit 対称

$$X(2\pi(\bar{j}+2^n)^*) = -X(2\pi(j+2^n)^*)$$

ならば, FFT アルゴリズムは対称

$$X^l(k+2^l \bar{j}) = -\bar{X}^l(k+2^l j)$$

である。

j が偶数のとき $\bar{j}$ は奇数であるから, いずれか一方の場合を求めればよい。

入力 $\bar{x}$ が, 実数かつ Hermit 対称ならば cosine 変換, 実数かつ Hermit 対称の場合は sine 変換となる。

定理の証明. FFT アルゴリズムの定義と仮定から

$$\begin{aligned} X^l(k+2^l \bar{j}) &= \sum_{0 \leq p < 2^l} X(2\pi(\bar{p}+2^l \bar{j}+2^n)^*) \cdot \exp(-2\pi i k \\ &\quad \cdot (p+2^l \bar{j}+2^n)^*) \\ &= \sum_{0 \leq p < 2^l} X(2\pi(2^{n-l}-p-2^l j+2^n)^*) \exp(-2\pi i k \\ &\quad \cdot (2^{n-l}-p-2^l j+2^n)^*) \\ &= \sum_{0 \leq p < 2^l} \bar{X}(2\pi(p+2^l j+2^n)^*) \exp(2\pi i k(p+ \end{aligned}$$

$$2^l j + 2^n)^*) = \bar{X}^l (k + 2^l j)$$

(証明終)

実数データに対するFFTにおいて、この対称性を利用する。jは偶数だからよりから n-l-1 ビットの整数となり

$$X^l \begin{pmatrix} k & 0 & 2j \\ \tilde{k} & 1 & 2j \end{pmatrix} \equiv X^l \begin{pmatrix} k & 0 & 0 & j \\ \tilde{k} & 1 & 0 & j \end{pmatrix}$$

すなわち l+1 ビット目は、いつも0である。

$$X^{l-1} \begin{pmatrix} k & 0 & 2j \\ \tilde{k} & 0 & 2j \end{pmatrix} \equiv X^{l-1} \begin{pmatrix} k & 0 & 0 & j \\ \tilde{k} & 0 & 0 & j \end{pmatrix}$$

$$\begin{aligned} X^{l-1} \begin{pmatrix} k & 1 & 2j \\ \tilde{k} & 1 & 2j \end{pmatrix} &\equiv X^{l-1} \begin{pmatrix} k & 1 & 0 & j \\ \tilde{k} & 1 & 0 & j \end{pmatrix} \\ &= \bar{X}^{l-1} \begin{pmatrix} k & 0 & 1 & \bar{j} \\ \tilde{k} & 0 & 1 & \bar{j} \end{pmatrix} \end{aligned}$$

最後の等式は、FFTテーブルの対称性に基づく。X^{l-1}(\cdot) の l ビット目は、やはり0である。

よって X^l(\cdot) の () 内の l+1 番目以降を左にうつめて0を除く。また n を n+1 にあらためる。したがって

$$X^l \begin{pmatrix} k & 0 & j \\ \tilde{k} & 1 & j \end{pmatrix}$$

において, 再び k , $\tilde{k}$ は $l-1$ ビット, j は $n-l$ ビットとなる. 重ね書きの cosine 変換の算法を導くために, j をあらためて $n-l-1$ ビットの j と $\bar{j} = 2^{n-l-1} - j$ に分離する.

以上のことに注意して次の算法が導かれる.

中点公式による高速 cosine 変換

$$X^l \begin{pmatrix} k & 0 & j \\ \tilde{k} & 1 & j \end{pmatrix} = X^{l-1} \begin{pmatrix} k & 0 & j \\ \tilde{k} & 0 & j \end{pmatrix} + \bar{X}^{l-1} \begin{pmatrix} k & 1 & \bar{j} \\ \tilde{k} & 1 & \bar{j} \end{pmatrix}$$

$$X^l \begin{pmatrix} \tilde{k} & 0 & j \\ k & 1 & j \end{pmatrix} = W \left\{ \bar{X}^{l-1} \begin{pmatrix} k & 0 & j \\ \tilde{k} & 0 & j \end{pmatrix} - X^{l-1} \begin{pmatrix} k & 1 & \bar{j} \\ \tilde{k} & 1 & \bar{j} \end{pmatrix} \right\}$$

$$X^l \begin{pmatrix} k & 0 & \bar{j} \\ \tilde{k} & 1 & \bar{j} \end{pmatrix} = X^{l-1} \begin{pmatrix} k & 0 & \bar{j} \\ \tilde{k} & 0 & \bar{j} \end{pmatrix} + \bar{X}^{l-1} \begin{pmatrix} k & 1 & j \\ \tilde{k} & 1 & j \end{pmatrix}$$

$$X^l \begin{pmatrix} k & 0 & \bar{j} \\ k & 1 & \bar{j} \end{pmatrix} = (-iW) \left\{ X^{l-1} \begin{pmatrix} k & 0 & \bar{j} \\ \tilde{k} & 0 & \bar{j} \end{pmatrix} - \bar{X}^{l-1} \begin{pmatrix} k & 1 & j \\ \tilde{k} & 1 & j \end{pmatrix} \right\}$$

$$\text{ただし } \bar{j} = 2^{n-l-1} - j, \quad \tilde{k} = 2^{l-1} - k \quad (7)$$

$$W = \exp\left(\frac{\pi}{2} i (j + 2^{n-l})^*\right)$$

特殊であるが, $j=0, 2^{n-l-2}, k=0, 2^{l-2}$ の場合の式は省略した.

2^n 項の cosine 変換に必要な三角関数表は

$$\exp\left(\frac{\pi}{4} \lambda j^*\right), \quad 0 \leq j < 2^{n-1}$$

でよい。なぜなら

$$\exp\left(\frac{\pi}{2} \lambda (2j + 2^{n-1})^*\right) = \exp\left(\frac{\pi}{4} \lambda (j + 2^{n-1-1})^*\right)$$

$$\exp\left(\frac{\pi}{2} \lambda (2j + 1 + 2^{n-1})^*\right) = \lambda \exp\left(\frac{\pi}{4} \lambda (\bar{j} + 2^{n-1-1})^*\right)$$

$$0 \leq j < 2^{n-1-1}, \quad \bar{j} = 2^{n-1-1} - 1 - j$$

が成立するからである。

また初期値の設定は、次のようにする。

半周期から 2^n 個の標本

$$x\left(\frac{\pi}{2^n} \left(j + \frac{1}{2}\right)\right), \quad 0 \leq j < 2^n$$

をつくり、2進逆順に並べ換えて FFT の算法 (7) の入力とする。

$x(t)$ が実数値の場合、FFT テーブルは共対称となるのでこの性質を用い、sine 変換に対する FFT を同様にして導くことができる。

参考文献

- 1) 島居達生：高速 sine 変換, cosine 変換とその数値積分への応用, 情報処理, Vol. 15, pp. 670-679 (1974)
- 2) 島居達生：フーリエ変換サブルーチン・パッケージの作成 (その 1, 2), 名古屋大学大型計算材料センターニュース, Vol. 10, NO1, NO2 (1979). (その 2) は印刷中.