第５章　対称アルゴリズム

入力データの実数性や対称性を用いて，計算量を減らすことができる．一つの方法は，それによる出力データの対称性を利用し，再帰計算の各段階で計算量を減らすことである．もう一つの方法は，入力・出力に変換を加え，少ない項数の複素DFTで計算する方法である．ここでは，後者を取り上げる．複素DFTの計算には，Cooley-TukeyのFFTを用い，項数nは2の巾とする．

◎部分ベクトル，部分行列のセミコロン記法

△ ベクトル$\boldsymbol{x}=(x_0,x_1,\cdots,x_{n-1})^{\mathrm{T}}\in\mathbb{R}^n$に対して，$\boldsymbol{x}_{p;h;k}\coloneqq(x_p,x_{p+h},\cdots,x_{p+(k-1)h})^{\mathrm{T}}\in\mathbb{R}^k$．

ここで，pは初項，hはステップ，kは次元を表す．特にステップ$h=1$のときはそれを省略して，

$\boldsymbol{x}_{p;k}\coloneqq(x_p,x_{p+1},\cdots,x_{p+(k-1)})^{\mathrm{T}}\in\mathbb{R}^k$と書くことがある．

同じく，行列についても，

△ 行列$A=(a_{i,j})\in\mathbb{R}^{m\times n}$に対して，$A_{p;s;k,q;t;l}\coloneqq(a_{p+is,q+jt})_{0\le i<k,0\le j<l}\in\mathbb{R}^{k\times l}$．

５．１　実FFT

$\boldsymbol{f}=(f_0,f_1,\cdots,f_{n-1})^{\mathrm{T}}\in\mathbf{C}^n$のDFT，$\boldsymbol{c}=(c_0,c_1,\cdots,c_{n-1})^{\mathrm{T}}=W_n\boldsymbol{f}$は

$$c_k=\sum_{l=0}^{n-1}\omega_n^{kl}f_l\ \ (0\le k<n),\omega_n=\exp\left(-\frac{2\pi i}{n}\right)\tag{1.1}$$

で与えられる．ベクトル$\boldsymbol{f},\boldsymbol{c}$の要素を周期$n$で

$$f_{jn+k}=f_k,\ c_{jn+k}=c_k\quad(-\infty<j<\infty,0\le k<n)\tag{1.2}$$

と拡張する．これにより，

$$c_k=\sum_{l=j}^{n+j-1}\omega_n^{kl}f_l\ \ (-\infty<k,j<\infty)\tag{1.3}$$

となることは，１章で示した．

入力$\boldsymbol{f}$が実ならそのデータ量は実質的に半分となる．ことのとき，出力$\boldsymbol{c}$の要素は共役対称

$$c_{n-k}=\overline{c}_k\tag{1.4}$$

となる．$n=2n',c_k=(a_k-ib_k)/2$とすると，

$$c_0=\frac{a_0}{2},c_{n'}=\frac{a_{n'}}{2},c_k=\frac{a_k-ib_k}{2},c_{n-k}=\frac{a_k+ib_k}{2}\ \ (0<k<n')\tag{1.5}$$

で，出力 $\boldsymbol{c}$ も n 個の実データ $a_k(0\le k\le n'),b_k(0<k<n')$ で記述される．したがって，その部分ベクトル $\boldsymbol{c}_{0;n'+1}$ が計算されれば十分である．

◎偶奇分け・高低分け

$n=2n'$ 項DFTに対するCooley-Tukey算法は，

$$\begin{aligned} c_k &= \sum_{l=0}^{n-1}\omega_n^{kl}f_l = \sum_{l=0}^{n'-1}\omega_n^{k(2l)}f_{2l} + \sum_{l=0}^{n'-1}\omega_n^{k(2l+1)}f_{2l+1} \\ &= \sum_{l=0}^{n'-1}\omega_{n'}^{kl}f_{2l} + \omega_n^k\sum_{l=0}^{n'-1}\omega_{n'}^{kl}f_{2l+1} = c_k^{(0)} + \omega_n^k c_k^{(1)} \end{aligned} \tag{1.6}$$

である．ここで，

$$\begin{aligned} \boldsymbol{c}^{(0)} &= (c_0^{(0)},c_1^{(0)},\cdots,c_{n'-1}^{(0)})^{\mathrm{T}} = W_{n'}\boldsymbol{f}_{0;2;n'}, \\ \boldsymbol{c}^{(1)} &= (c_0^{(1)},c_1^{(1)},\cdots,c_{n'-1}^{(1)})^{\mathrm{T}} = W_{n'}\boldsymbol{f}_{1;2;n'} \end{aligned} \tag{1.7}$$

は周期 n' であるから，

$$\tilde{\boldsymbol{c}}^{(0)} = \begin{pmatrix} \boldsymbol{c}^{(0)} \\ c_0^{(0)} \end{pmatrix},\ \tilde{\boldsymbol{c}}^{(1)} = \begin{pmatrix} \boldsymbol{c}^{(1)} \\ c_0^{(1)} \end{pmatrix} \tag{1.8}$$

とすると，式(1.6)より， $\boldsymbol{c}_{0;n'+1}$ は

$$\begin{aligned} \boldsymbol{c}_{0;n'+1} &= \tilde{\boldsymbol{c}}^{(0)} + \boldsymbol{\Omega}_n^{n'}\tilde{\boldsymbol{c}}^{(1)}, \\ \boldsymbol{\Omega}_n^{n'} &= \mathrm{diag}(1,\omega_n,\omega_n^2,\cdots,\omega_n^{n'}) = \mathrm{diag}(1,\omega_n,\omega_n^2,\cdots,\omega_n^{n'-1},-1) \end{aligned} \tag{1.9}$$

で計算できる．

◎混合計算と再生

入力 $\boldsymbol{f}$ は実数であるから， $\boldsymbol{c}^{(0)},\boldsymbol{c}^{(1)}$ も共役対称：

$$c_{n'-k}^{(0)} = \overline{c_k^{(0)}},\ c_{n'-k}^{(1)} = \overline{c_k^{(1)}}$$

である．ゆえに， $\boldsymbol{c}^{(0)},\boldsymbol{c}^{(1)}$ の混合計算

$$\boldsymbol{d} = (d_0,\cdots,d_{n'-1})^{\mathrm{T}} = W_{n'}(\boldsymbol{f}_{0;2;n'} + i\boldsymbol{f}_{1;2;n'}) = \boldsymbol{c}^{(0)} + i\boldsymbol{c}^{(1)} \tag{1.10}$$

から， $\boldsymbol{c}^{(0)},\boldsymbol{c}^{(1)}$ が次のようにして再生される．

$$\begin{aligned} \boldsymbol{c}^{(0)} &= \frac{1}{2}(\boldsymbol{d} + \overline{\boldsymbol{d}_{n';-1;n'}}), \\ \boldsymbol{c}^{(1)} &= \frac{1}{2i}(\boldsymbol{d} - \overline{\boldsymbol{d}_{n';-1;n'}}). \end{aligned} \tag{1.11}$$

なぜなら，

$$d_k = c_k^{(0)} + ic_k^{(1)},$$
$$\overline{d_{n'-k}} = \overline{c_{n'-k}^{(0)}} - i\overline{c_{n'-k}^{(1)}} = c_k^{(0)} - ic_k^{(1)}$$

を $c_k^{(0)}, c_k^{(1)}$ について解いて，

$$c_k^{(0)} = \frac{1}{2}(d_k + \overline{d_{n'-k}}),$$
$$c_k^{(1)} = \frac{1}{2i}(d_k - \overline{d_{n'-k}}).$$

だからである．

◎アルゴリズム

式(1.10), (1.11), (1.9)をまとめて，$\boldsymbol{f}$ から $\boldsymbol{c}_{0;n'+1}$ を計算する実FFTの算法を得る．

＜n 項実FFT＞

1．n' 項複素FFT　：$\boldsymbol{d} = (d_0, \cdots, d_{n'-1})^{\mathrm{T}} = W_{n'}(\boldsymbol{f}_{0;2;n'} + i\boldsymbol{f}_{1;2;n'})$

2．$\boldsymbol{c}^{(0)}, \boldsymbol{c}^{(1)}$ の再生：$\boldsymbol{c}^{(0)} = \frac{1}{2}(\boldsymbol{d} + \overline{\boldsymbol{d}_{n';-1;n'}}), \boldsymbol{c}^{(1)} = \frac{1}{2i}(\boldsymbol{d} - \overline{\boldsymbol{d}_{n';-1;n'}}).$

3．合成　　：$\boldsymbol{c}_{0;n'+1} = \tilde{\boldsymbol{c}}^{(0)} + \boldsymbol{\Omega}_n^{n'}\tilde{\boldsymbol{c}}^{(1)}.$

◎計算量

n 項複素FFTの乗算数を $Cn\log_2 n + O(n)$ とすれば，n 項実FFTの乗算数は

$$Cn'\log_2 n' + O(n) = \frac{C}{2}n\log_2 n + O(n) \qquad (1.12)$$

で元の計算量の約半分である．

5．2　実半対称FFT

実入力が半対称 $f_{-l} = f_{l-1}$ $(-\infty < l < \infty)$ のとき，式(1.7)より，

$$c_k^{(1)} = \sum_{l=0}^{n'-1}\omega_{n'}^{kl}f_{2l+1} = \sum_{l=0}^{n'-1}\omega_{n'}^{kl}f_{-2l-2} = \omega_{n'}^{-k}\sum_{l=0}^{n'-1}\omega_{n'}^{k(l+1)}f_{-2(l+1)}$$
$$= \omega_{n'}^{-k}\overline{\sum_{l=0}^{n'-1}\omega_{n'}^{-k(l+1)}f_{-2(l+1)}} = \overline{\omega_{n'}^{k}c_k^{(0)}}.$$

すなわち，n' 項実FFTで $\boldsymbol{c}^{(0)}$ のみを計算しておけば，$\boldsymbol{c}^{(1)}$ は

$$\boldsymbol{c}^{(1)} = \overline{\boldsymbol{\Omega}_{n'}^{n'-1}\boldsymbol{c}^{(0)}},$$
$$\boldsymbol{\Omega}_{n'}^{n'-1} = \mathrm{diag}(1, \omega_{n'}, \omega_{n'}^2, \cdots, \omega_{n'}^{n'-1})$$

で得られる．$c_{n'}^{(0)} = \sum_{l=0}^{n'-1} f_{2l} \in \mathbb{R}$ より，$c_{n'}^{(1)} = \overline{\omega_{n'}^{n'} c_{n'}^{(0)}} = c_{n'}^{(0)}$ ゆえ，式(1.9)より，$c_{n'} = c_{n'}^{(0)} - c_{n'}^{(1)} = 0$ であることに注意する．以上より次のアルゴリズムを得る．

＜n 項実半対称FFT＞（n' 項実FFTを1回用いる）

1．n' 項実FFT ：$\boldsymbol{c}^{(0)} = W_{n'} \boldsymbol{f}_{0;2;n'}$．

2．$\boldsymbol{c}^{(1)}$ の構成 ：$\boldsymbol{c}^{(1)} = \overline{\Omega_{n'}^{n'-1} \boldsymbol{c}^{(0)}}$．

3．合成 ：$\boldsymbol{c}_{0;n'+1} = \begin{pmatrix} \boldsymbol{c}^{(0)} + \Omega_n^{n'-1} \boldsymbol{c}^{(1)} \\ 0 \end{pmatrix}$．

総乗算数は n 項実DFT(1.12)からさらに半減して，

$$\frac{C}{4} n \log_2 n + O(n) \tag{2.1}$$

となる．

５．３　実対称FFT（台形則高速cosine変換）

実入力 $\boldsymbol{f} = (f_0, f_1, \cdots, f_{n-1})^{\mathrm{T}} \in \mathbf{R}^n$ が対称 $f_{-l} = f_l$ $(-\infty < l < \infty)$ のとき，出力 $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})^{\mathrm{T}} = W_n \boldsymbol{f}$ もまた実かつ対称 $c_{-k} = c_k$ $(-\infty < k < \infty)$ であることは1章で示した．虚部 $(= 0)$ を計算する必要はなく計算量は n 項実DFT(1.12)からさらに半減することが期待できる．

◎再帰的計算法

対称性より，

$$f_{2(-l)} = f_{2l},$$
$$f_{2(-l)+1} = f_{2l-1} = f_{2(l-1)+1}$$

ゆえ，$\boldsymbol{f}_{0;2;n'}$ は対称，$\boldsymbol{f}_{1;2;n'}$ は半対称である．したがって，式(1.7)の $\boldsymbol{c}^{(0)}$ は n' 項実対称DFT, $\boldsymbol{c}^{(1)}$ は n' 項実半対称DFTである．これらを，式(1.9)で合成すれば，求める $\boldsymbol{c}$ が得られる．すなわち，

$$n \text{項実対称DFT} = \frac{n}{2} \text{項実対称DFT} + \frac{n}{2} \text{項実半対称DFT} + \text{合成(1.9)} \tag{3.1}$$

と言う図式となる．これを $n/2$ 項実対称DFTに再帰的に用いて実対称FFTを得る．

◎計算量

［主張3.1］n 項実半対称FFTの乗算数を

$$\mu_n^{(1)} \le \frac{C}{4} n \log_2 n + Dn$$

式(1.9)の合成に要する乗算数を

$$\mu_n^{(2)} \le Bn$$

とすると，式(3.1)を再帰的に用いたn項実対称DFTの乗算数

$$\mu_n \le \frac{C}{4} n \log_2 n + (D+2B)n \tag{3.2}$$

である．

（証明）1項DFT $c_0 = f_0$ は計算を要しない． $n = 2^m$ 項実対称DFTを計算するには， $2,4,\cdots,2^{m-1}$項実半対称FFTと， $2,4,\cdots,2^m$ 項の合成である．ゆえに，

$$\begin{aligned}\mu_n &\le \sum_{l=1}^{m-1}\left(\frac{C}{4}2^l l + D2^l\right) + \sum_{l=1}^{m} B2^l \le \frac{C}{4}\left(2^m m - 2^{m+1} + 2\right) + Dn + 2Bn \\ &\le \frac{C}{4} n \log_2 n + (D+2B)n\end{aligned}$$

となる．／／

５．４　実半歪対称FFT

実入力が半歪対称 $f_{-l} = -f_{l-1}$ $(-\infty < l < \infty)$ のとき，式(1.7)より，

$$\begin{aligned}c_k^{(1)} &= \sum_{l=0}^{n'-1} \omega_{n'}^{kl} f_{2l+1} = -\sum_{l=0}^{n'-1} \omega_{n'}^{kl} f_{-2l-2} = -\omega_{n'}^{-k} \sum_{l=0}^{n'-1} \omega_{n'}^{k(l+1)} f_{-2(l+1)} \\ &= -\omega_{n'}^{-k} \overline{\sum_{l=0}^{n'-1} \omega_{n'}^{-k(l+1)} f_{-2(l+1)}} = -\overline{\omega_{n'}^{k} c_k^{(0)}}\end{aligned}.$$

すなわち， n'項実FFTで $\boldsymbol{c}^{(0)}$ のみを計算しておけば， $\boldsymbol{c}^{(1)}$ は

$$\boldsymbol{c}^{(1)} = -\overline{\Omega_{n'}^{n'-1} \boldsymbol{c}^{(0)}}$$

で得られる． $c_{n'}^{(0)} = \sum_{l=0}^{n'-1} f_{2l} \in \mathbb{R}$ より， $c_{n'}^{(1)} = -\overline{\omega_{n'}^{n'} c_{n'}^{(0)}} = -c_{n'}^{(0)}$ ゆえ，式(1.9)より， $c_{n'} = c_{n'}^{(0)} - c_{n'}^{(1)} = 2c_{n'}^{(0)}$ であることに注意する．以上より次のアルゴリズムを得る．以上より次のアルゴリズムを得る．

＜n項実半歪対称FFT＞（n'項実FFTを1回用いる）

1．n'項実FFT　：$\boldsymbol{c}^{(0)} = W_{n'} \boldsymbol{f}_{0;2;n'}$．

2．$\boldsymbol{c}^{(1)}$の構成　：$\boldsymbol{c}^{(1)} = -\overline{\Omega_{n'}^{n'-1} \boldsymbol{c}^{(0)}}$．

3．合成　　：$\boldsymbol{c}_{0;n'+1} = \begin{pmatrix} \boldsymbol{c}^{(0)} + \Omega_n^{n'-1} \boldsymbol{c}^{(1)} \\ 2c_{n'}^{(0)} \end{pmatrix}$．

総乗算数は項実DFT(1.12)からさらに半減して，

$$\frac{C}{4} n \log_2 n + O(n) \tag{4.1}$$

となる．

５．５　実歪対称FFT（台形則高速sine変換）

入力 $\boldsymbol{f} = (f_0, f_1, \cdots, f_{n-1})^{\mathrm{T}} \in \mathbf{R}^n$ が歪対称 $f_{-l} = -f_l$ $(-\infty < l < \infty)$ のとき，出力 $\boldsymbol{c} = (c_0, c_1, \cdots, c_{n-1})^{\mathrm{T}} = W_n \boldsymbol{f}$ は純虚数で歪対称 $c_{-k} = -c_k$ $(-\infty < k < \infty)$ であることは1章で述べた．実部を計算する必要はなく計算量は項実DFT(1.12)からさらに半減することが期待できる．

◎再帰的計算法

歪対称性より，

$$f_{2(-l)} = -f_{2l},$$
$$f_{2(-l)+1} = -f_{2l-1} = f_{2(l-1)+1}$$

ゆえ，$\boldsymbol{f}_{0;2;n'}$ は歪対称，は半歪対称である．したがって，式(1.7)のは n' 項実歪対称DFT,は n' 項実半歪対称DFTである．これらを，式(1.9)で合成すれば，求めるが得られる．すなわち，

$$\text{項実歪対称DFT} = \frac{n}{2}\text{項実歪対称DFT} + \frac{n}{2}\text{項実半歪対称DFT} + \text{合成(1.9)} \tag{5.1}$$

と言う図式となる．これを $n/2$ 項実歪対称DFTに再帰的に用いて実歪対称FFTを得る．

◎計算量

［主張4.1］項実半歪対称FFTの乗算数を

$$\mu_n^{(1)} \le \frac{C}{4} n \log_2 n + Dn$$

式(1.9)の合成に要する乗算数を

$$\mu_n^{(2)} \le Bn$$

とすると，式(5.1)を再帰的に用いた項実歪対称DFTの乗算数

$$\mu_n \le \frac{C}{4} n \log_2 n + (D + 2B)n \tag{5.2}$$

である．

（証明）主張3.1の証明と同じ．／／