# 第４章　Winogradのアルゴリズム

WinogradのアルゴリズムはGoodのアルゴリズムを基礎としてその乗算回数が減少するように改良したものである．加算数はかなり増加するので，現在のパソコンのCPUのように，乗算と加算のクロック数がほぼ同じ計算機では不利となる．その対策として，Winogradの算法の加算数を減らすSplit-Algorithmがある．

## ４.１　Goodのアルゴリズムの記述

Winogradのアルゴリズムを説明するため，Goodのアルゴリズムの主要部を簡潔に記述する．$n = p_1 n_2$ 項のGoodのアルゴリズムは

$$c_{\varphi_1(k_1,k)} = \sum_{l_1=0}^{p_1-1} w_{p_1}^{k_1 l_1} \sum_{l=0}^{n_2-1} w_{n_2}^{kl} f_{\psi_1(l_1,l)} \qquad (0 \le k_1 < p_1, 0 \le k < n_2)$$

である．$n_2 = p_2 p_3$ と因数分解されるとき，内側の総和にGoodのアルゴリズムを用いて，

$$c_{\varphi_1(k_1,\varphi_2(k_2,k_3))} = \sum_{l_1=0}^{p_1-1} w_{p_1}^{k_1 l_1} \sum_{l_2=0}^{p_2-1} w_{p_2}^{k_2 l_2} \sum_{l_3=0}^{p_3-1} w_{p_3}^{k_3 l_3} f_{\psi_1(l_1,\psi_2(l_2,l_3))} \qquad (0 \le k_1 < p_1, 0 \le k_2 < p_2, 0 \le k_3 < n_3)$$

である．ここで，3次元配列を

$$\boldsymbol{c} = (c_{k_1,k_2,k_3} = c_{\varphi_1(k_1,\varphi_2(k_2,k_3))})_{0 \le k_1 < p_1, 0 \le k_2 < p_2, 0 \le k_3 < p_3} \in \mathbb{R}^{p_1 \times p_2 \times p_3},$$
$$\boldsymbol{f} = (f_{l_1,l_2,l_3} = f_{\psi_1(l_1,\psi_2(l_2,l_3))})_{0 \le l_1 < p_1, 0 \le l_2 < p_2, 0 \le l_3 < p_3} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$$

と定義すると，Goodのアルゴリズムの主要部（データの再配置を除いた部分）は

$$c_{k_1,k_2,k_3} = \sum_{l_1=0}^{p_1-1} w_{p_1}^{k_1 l_1} \sum_{l_2=0}^{p_2-1} w_{n_2}^{kl} \sum_{l=0}^{p_3-1} w_{p_3}^{kl} f_{l_1,l_2,l_3} \qquad (0 \le k_1 < p_1, 0 \le k_2 < p_2, 0 \le k < p_3)$$

と書ける．この操作を再帰的に行えば一般に，項数 $n = p_1 p_2 \cdots p_m$ のGoodのアルゴリズムの主要部は $m$ 次元配列 $\boldsymbol{c}, \boldsymbol{f} \in \mathbb{C}^{p_1 \times \cdots \times p_m}$ に関する変換

$$c_{k_1,\cdots,k_m} = \sum_{l_1=0}^{p_1-1} w_{p_1}^{k_1 l_1} \sum_{l_2=0}^{p_2-1} \cdots \sum_{l_m=0}^{p_m-1} w_{p_m}^{k_m l_m} f_{l_1,\cdots,l_m} \qquad (0 \le k_i < p_i \ (1 \le i \le m)) \tag{1.1}$$

である．

◎各次元変換

$m$ 次元配列 $\boldsymbol{c}, \boldsymbol{f} \in \mathbb{C}^{p_1 \times \cdots \times p_m}$ の添え字集合を

$$I(p_1,\cdots,p_m) = \{(k_1,\cdots,k_m) : 0 \le k_i < p_i \ (1 \le i \le m)\}$$

と書く．

行列 $A = (a_{kl}) \in \mathbb{C}^{q_i \times p_i}$ による $\boldsymbol{x} \in \mathbb{C}^{p_1 \times \cdots \times p_i \times \cdots \times p_m}$ から $\boldsymbol{y} \in \mathbb{C}^{p_1 \times \cdots \times q_i \times \cdots \times p_m}$ への第 $i$ 次元線形変換 $\boldsymbol{y} = A^{(i)}\boldsymbol{x}$ を

$$y_{l_1 \cdots k_i \cdots l_m} = \sum_{l_i=0}^{p_i-1} a_{k_i l_i} x_{l_1 \cdots l_i \cdots l_m} \ , (l_1 \cdots k_i \cdots l_m) \in I(p_1,\cdots,q_i,\cdots,p_m) \tag{1.2}$$

で定義する．$\boldsymbol{x}$ から $\boldsymbol{y}$ へ，第 $i$ 次元のサイズのみが $p_i$ から $q_i$ に変化する．

配列 $\boldsymbol{f}\in\mathbb{C}^{p_1\times\cdots\times p_m}$ の部分配列 $\boldsymbol{f}_{l_1,\cdots,l_{i-1},*,l_{i+1},\cdots,l_m}\ \left((l_1,\cdots,l_{i-1},l_{i+1},\cdots,l_m)\in I(p_1,\cdots,p_{i-1},p_{i+1},\cdots,p_m)\right)$ を

$$\boldsymbol{f}_{l_1,\cdots,l_{i-1},*,l_{i+1},\cdots,l_m}=\left(f_{l_1,\cdots,l_{i-1},l_i,l_{i+1},\cdots,l_m}\right)_{l_i=0}^{p_i-1} \tag{1.3}$$

で定義する．これを用いて(1.2)は

$$\boldsymbol{c}_{l_1,\cdots,l_{i-1},*,l_{i+1},\cdots,l_m}=A\boldsymbol{f}_{l_1,\cdots,l_{i-1},*,l_{i+1},\cdots,l_m}\ \left((l_1,\cdots,l_{i-1},l_{i+1},\cdots,l_m)\in I(p_1,\cdots,p_{i-1},p_{i+1},\cdots,p_m)\right) \tag{1.4}$$

と書ける．

［主張1.1］ $A\in\mathbb{C}^{q_i\times p_i},B\in\mathbb{C}^{q_j\times p_j}\ (i\neq j)$ なら， $A^{(i)}B^{(j)}=B^{(j)}A^{(i)}$ である．//

（証明） $\boldsymbol{y}=A^{(i)}B^{(j)}\boldsymbol{x},\boldsymbol{z}=B^{(j)}A^{(i)}$ とすると， $\boldsymbol{y},\boldsymbol{z}\in\mathbb{C}^{p_1\times\cdots\times q_i\times\cdots\times q_j\times\cdots\times p_m}$ であり，

$$y_{l_1\cdots k_i\cdots k_j\cdots l_m}=\sum_{l_i=0}^{p_i-1}a_{k_il_i}\left(\sum_{l_j=0}^{p_j-1}b_{k_jl_j}x_{l_1\cdots l_i\cdots l_j\cdots l_m}\right)=\sum_{l_j=0}^{p_j-1}b_{k_jl_j}\left(\sum_{l_i=0}^{p_i-1}a_{k_il_i}x_{l_1\cdots l_i\cdots l_j\cdots l_m}\right)=z_{l_1\cdots k_i\cdots k_j\cdots l_m}$$

が全ての $(l_1\cdots k_i\cdots k_j\cdots l_m)\in I(p_1,\cdots,q_i,\cdots,q_j,\cdots,p_m)$ について成立する．//

各次元変換を用いて，項数 $n=p_1p_2\cdots p_m$ のGoodのアルゴリズムの主要部(1.1)は

$$\boldsymbol{c}=G_n\boldsymbol{f}=W_{p_1}^{(1)}W_{p_2}^{(2)}\cdots W_{p_m}^{(m)}\boldsymbol{f}\quad,\boldsymbol{f}\in\mathbb{C}^{p_1\times\cdots\times p_m} \tag{1.5}$$

と書ける．

## 4.2　小DFTの分割

資料「Small DFTs」に $p\in\{2,3,4,5,7,8,9,16\}$ の小DFTのアルゴリズムを集めた．これらを用いて項数

$$n=2^i3^j5^k7^l\ \ (0\le i\le 4,0\le j\le 2,0\le k\le 1,0\le l\le 1), \tag{2.1}$$

すなわち， $n=2,3,\cdots,5040$ のDFTをGoodのアルゴリズムで構成できる．

Winogradのアルゴリズムでは，上記小DFT $W_p$ のアルゴリズムを次のように3つの因子に分解する．

$$W_p=A_pM_pB_p \tag{2.2}$$

$B_p\in\mathbf{C}^{q\times p}$ ：前加算(addition Before multiplication)．複素加算と虚数単位の乗算で構成．

$M_p\in\mathbf{C}^{q\times q}$ ：中乗算(Multiplication)．実定数×複素乗算．対角行列．

$A_p\in\mathbf{C}^{p\times q}$ ：後加算(addition After multiplication)．複素加算で構成される部分．

これをWinograd分解という． $W_p$ は正則ゆえ $q\ge p$ である．

資料では，変数 $m_i(0\le i<q)$ で中乗算の結果を受けている．変数 $m_i$ の乗算に先立つ加算が前加算である．変数 $m_i(0\le i<q)$ から出力 $\bar{X}_i(0\le i<p)$ への変換部が後加算である．

注意事項：

1. 虚数単位i（資料では電気屋の $j$）の乗算は，前加算に含まれる．計算機では虚部符号反転と実部虚部の交換に還元されるからである．

2. 中乗算には1の乗算が含まれる．資料ではその回数を（）内に示している．

## 4.3　Winogradのアルゴリズム

項数が$n=p_1p_2$と互いに素な因子$p_1,p_2$に分解されているとする．Goodの算法の計算の主要部は

$$G_n = W_{p_1}^{(1)} W_{p_2}^{(2)} \tag{3.1}$$

である．各小DFTをWinograd分解した$W_{p_i}=A_{p_i}M_{p_i}B_{p_i}$ $(i=1,2)$を代入して，主張1.1を用いることにより，$G_n$のWinograd分解

$$\begin{aligned} G_n &= (A_{p_1}M_{p_1}B_{p_1})^{(1)}(A_{p_2}M_{p_2}B_{p_2})^{(2)}\cdots(A_{p_m}M_{p_m}B_{p_m})^{(m)} \\ &= A_{p_1}^{(1)}M_{p_1}^{(1)}B_{p_1}^{(1)}A_{p_2}^{(2)}M_{p_2}^{(2)}B_{p_2}^{(2)}\cdots A_{p_m}^{(m)}M_{p_m}^{(m)}B_{p_m}^{(m)} \\ &= A_{p_1}^{(1)}A_{p_2}^{(2)}\cdots A_{p_m}^{(m)}M_{p_1}^{(1)}M_{p_2}^{(2)}\cdots M_{p_m}^{(m)}B_{p_1}^{(1)}B_{p_2}^{(2)}\cdots B_{p_m}^{(m)} = A_nM_nB_n \end{aligned} \tag{3.2}$$

を得る．ここで，

$$\begin{aligned} A_n &= A_{p_1}^{(1)}A_{p_2}^{(2)}\cdots A_{p_m}^{(m)}, \\ M_n &= M_{p_2}^{(2)}\cdots M_{p_m}^{(m)}, \\ B_n &= B_{p_1}^{(1)}B_{p_2}^{(2)}\cdots B_{p_m}^{(m)} \end{aligned} \tag{3.3}$$

であり，それぞれ$G_n$の後加算，中乗算，前加算と呼ぶ．後加算$A_n$と前加算$B_n$は加法のみで計算でき，計算量は次の主張となる．

［主張3.1］$B_{p_i}$の加算数を$\alpha_B(p_i)$とすると，前加算$B_n$に要する加算数は

$$\alpha_B(p_1,p_2,\cdots,p_m) = n\sum_{i=1}^{m}\frac{\alpha_B(p_i)}{p_i}\prod_{j=i+1}^{m}\frac{q_j}{p_j}. \tag{3.4}$$

$A_{p_i}$の加算数を$\alpha_A(p_i)$とすると，後加算に要する加算数$A_n$は

$$\alpha_A(p_1,p_2,\cdots,p_m) = n\sum_{i=1}^{m}\frac{\alpha_A(p_i)}{p_i}\prod_{j=1}^{i-1}\frac{q_j}{p_j}. \quad // \tag{3.5}$$

（証明）前加算のみを示す．後加算についても同様である．$\boldsymbol{f}^{(i)}\in\mathbb{C}^{p_1\times\cdots\times p_i\times q_{i+1}\times\cdots\times q_m}$として，前加算の計算手続きは

$$\begin{aligned} &\boldsymbol{f}^{(m)} = \boldsymbol{f}, \\ &\boldsymbol{f}^{(i-1)} = B_{p_i}^{(i)}\boldsymbol{f}^{(i)} \quad (i=m,m-1,\cdots,1) \end{aligned}$$

である．結果として$\boldsymbol{f}^{(0)} = B_{p_1}^{(1)}B_{p_2}^{(2)}\cdots B_{p_m}^{(m)}\boldsymbol{f}^{(m)} = B_n\boldsymbol{f}$を得る．$\boldsymbol{f}^{(i-1)} = B_{p_i}^{(i)}\boldsymbol{f}^{(i)}$を部分配列ごとに書き下すと

$$\boldsymbol{f}_{l_1,\cdots,l_{i-1},*,k_{i+1},\cdots,k_m}^{(i-1)} = B_{p_i}\boldsymbol{f}_{l_1,\cdots,l_{i-1},*,k_{i+1},\cdots,k_m}^{(i)} \quad ,(l_1,\cdots,l_{i-1},k_{i+1},\cdots,k_m)\in I(p_1,\cdots,p_{i-1},q_{i+1},\cdots,q_m)$$

となる．ゆえに，その加算数は

$$p_1\cdots p_{i-1}q_{i+1}\cdots q_m\alpha_B(p_i) = p_1\cdots p_{i-1}\left(p_i\frac{\alpha_B(p_i)}{p_i}\right)\left(p_{i+1}\cdots p_m\prod_{j=i+1}^{m}\frac{q_j}{p_j}\right) = \frac{n\alpha_B(p_i)}{p_i}\prod_{j=i+1}^{m}\frac{q_j}{p_j}$$

となる．これを$i$について総和して(3.4)を得る．//

中乗算$M_n$の計算量については主張を得る．

［主張3.2］中乗算$M_n$の作用に要するのは乗算のみで，その乗算（実×複素）数は，

$$\mu(p_1,p_2,\cdots,p_m)=\prod_{i=1}^{m}q_i\ .\ // \tag{3.6}$$

（証明） $\boldsymbol{f}^{(i)}\in\mathbb{C}^{q_1\times\cdots\times q_i\times q_{i+1}\times\cdots\times q_m}$ として，前加算の計算手続きは

$$\boldsymbol{f}^{(m)}=\boldsymbol{f},$$
$$\boldsymbol{f}^{(i-1)}=M_{p_i}^{(i)}\boldsymbol{f}^{(i)}\ \ (i=m,m-1,\cdots,1)$$

である．結果として$\boldsymbol{f}^{(0)}=M_{p_1}^{(1)}M_{p_2}^{(2)}\cdots M_{p_m}^{(m)}\boldsymbol{f}^{(m)}=M_n\boldsymbol{f}$を得る．$\boldsymbol{f}^{(i-1)}=M_{p_i}^{(i)}\boldsymbol{f}^{(i)}$を部分配列ごとに書き下すと，$M_{p_i}=\mathrm{diag}(\gamma_{i0},\gamma_{i1},\cdots,\gamma_{i,p_i-1})$として，

$$\boldsymbol{f}^{(i-1)}_{k_1,\cdots,k_{i-1},*,k_{i+1},\cdots,k_m}=M_{p_i}\boldsymbol{f}^{(i)}_{k_1,\cdots,k_{i-1},*,k_{i+1},\cdots,k_m}\ \ ,(k_1,\cdots,k_{i-1},k_{i+1},\cdots,k_m)\in I(q_1,\cdots,q_{i-1},q_{i+1},\cdots,q_m)\ .$$

これをさらに要素ごとに書き下せば，$M_{p_i}=\mathrm{diag}(\gamma_{i0},\gamma_{i1},\cdots,\gamma_{i,p_i-1})$として，

$$f^{(i-1)}_{k_1,\cdots,k_m}=\gamma_{p_ik_i}f^{(i)}_{k_1,\cdots,k_m}\ \ ,(k_1,\cdots,k_m)\in I(q_1,\cdots,q_m)\ .$$

ゆえに，

$$f^{(0)}_{k_1,\cdots,k_m}=\gamma_{p_1k_1}\cdots\gamma_{p_mk_m}f^{(i)}_{k_1,\cdots,k_m}\ \ ,(k_1,\cdots,k_m)\in I(q_1,\cdots,q_m)$$

となる．あらかじめ$\gamma_{k_1,\cdots,k_m}=\gamma_{p_1k_1}\cdots\gamma_{p_mk_m}\ \ (k_1,\cdots,k_m)\in I(q_1,\cdots,q_m)$を計算し，データとして保存しておけば，

$$f^{(0)}_{k_1,\cdots,k_m}=\gamma_{k_1,\cdots,k_m}f^{(i)}_{k_1,\cdots,k_m}\ \ ,(k_1,\cdots,k_m)\in I(q_1,\cdots,q_m)$$

を計算すればよいので，その乗算（実×複素）数は式(3.6)となる．//

［例］$n=p_1p_2p_3p_4=16\cdot9\cdot5\cdot7=5040$のとき，$q_1=18,q_2=11,q_3=6,q_4=9$ゆえ，

$$\mu(p_1,p_2,p_3,p_4)=18\cdot11\cdot6\cdot9=10692$$

である．複素×複素乗算に換算して5346回で，項数$n=5040$とほぼ等しい！Goodの算法では

$$n\left(\frac{\mu(p_1)}{p_1}+\frac{\mu(p_2)}{p_2}+\frac{\mu(p_3)}{p_3}+\frac{\mu(p_4)}{p_4}\right)=5040\left(\frac{10}{16}+\frac{10}{9}+\frac{5}{5}+\frac{8}{7}\right)=19550\ \text{回}$$

であるから，乗算数は半減した．加算数に関しては因子$\prod_{j=i+1}^{m}q_i/p_i$, $\prod_{j=1}^{i-1}q_i/p_i$のため，一般にGoodの算法の加算数より増加する．具体的には，Goodの182012実加算に対し233928実加算である．

## 4.4　加算数を最小化する計算順序

主張1.1より，前加算$B_n=B_{p_1}^{(1)}B_{p_2}^{(2)}\cdots B_{p_m}^{(m)}$の積の順序は自由である．すなわち，$\{1,2,\cdots,m\}$の順列を$\{\sigma(1),\sigma(2),\cdots,\sigma(m)\}$とするとき，

$$B_n = B_{p_{\sigma(1)}}^{(\sigma(1))} B_{p_{\sigma(2)}}^{(\sigma(2))} \cdots B_{p_{\sigma(m)}}^{(\sigma(m))}$$

である．この加算数

$$\alpha_B(p_{\sigma(1)}, p_{\sigma(2)}, \cdots, p_{\sigma(m)}) = n \sum_{i=1}^{m} \frac{\alpha_B(p_{\sigma(i)})}{p_{\sigma(i)}} \prod_{j=i+1}^{m} \frac{q_{\sigma(j)}}{p_{\sigma(j)}}$$

は順列$\sigma$によって異なる．それは，次のようにして最適化される．

［定理4.2］指標を

$$\rho(p_i) = \begin{cases} * & ,q_i = p_i,\ \alpha_B(p_i) = 0, \\ \infty & ,q_i = p_i,\ \alpha_B(p_i) \neq 0 \\ \alpha_B(p_i)/(q_i - p_i) & ,q_i \neq p_i \end{cases} \tag{4.1}$$

で定義する．$\rho(p_i) = *$の因子$B_{p_i}^{(i)}$はどこに配置してもよい．そうでない因子は指標で昇順

$$\rho(p_{\sigma(1)}) \leq \rho(p_{\sigma(2)}) \leq \cdots \leq \rho(p_{\sigma(m)}) \tag{4.2}$$

となるように並べ替えると$\alpha_B(p_{\sigma(1)}, p_{\sigma(2)}, \cdots, p_{\sigma(m)})$は最小である．//

（証明）レポートの課題とする．//

（ヒント：隣り合う因子の互換の効果を考える．バブル・ソート．）

同じく，

［定理4.3］式(4.1)の指標$\rho$で，の因子$A_{p_i}^{(i)}$はどこに配置してもよい．そうでない因子は指標で降順

$$\rho(p_{\sigma(1)}) \geq \rho(p_{\sigma(2)}) \geq \cdots \geq \rho(p_{\sigma(m)}) \tag{4.3}$$

となるように並べ替えると$\alpha_A(p_{\sigma(1)}, p_{\sigma(2)}, \cdots, p_{\sigma(m)})$は最小である．//

## 4.5　Split-Algorithm

前加算，後加算の加算数を減らす方法にSplit-Algorithmがある．前加算について説明する．後加算についても同様の処理を施す．小DFT$W_p$の前加算$B_p \in \mathbf{C}^{q \times p}$で，$q > p$のとき，

$$B_p = B_p'' B_p',\ B_p' \in \mathbf{C}^{p \times p},\ B_p'' \in \mathbf{C}^{q \times p}$$

と分解する．$B_p', B_p''$の加算数$\alpha_B'(p), \alpha_B''(p)$が$\alpha_B(p) = \alpha_B'(p) + \alpha_B''(p)$を満たし，なるべく$\alpha_B'(p)$の方が大きくなるように分解する．これにより，

$$B_n = B_{p_1}^{(1)} B_{p_2}^{(2)} \cdots B_{p_m}^{(m)} = \left( B_{p_1}''^{(1)} B_{p_2}''^{(2)} \cdots B_{p_m}''^{(m)} \right) \left( B_{p_1}'^{(1)} B_{p_2}'^{(2)} \cdots B_{p_m}'^{(m)} \right) = B_n'' B_n' \tag{5.1}$$

である．前節と同様にして$B_n', B_n''$の加算数はそれぞれ

$$\begin{aligned} \alpha_B'(p_1, p_2, \cdots, p_m) &= n \sum_{i=1}^{m} \frac{\alpha_B'(p_i)}{p_i}, \\ \alpha_B''(p_1, p_2, \cdots, p_m) &= n \sum_{i=1}^{m} \frac{\alpha_B''(p_i)}{p_i} \prod_{j=1}^{i-1} \frac{q_j}{p_j} \end{aligned} \tag{5.2}$$

となり，その合計は$\alpha_B(p_1, p_2, \cdots, p_m)$以下である．具体的な改善具合は，次ページの表を見よ．