応用数学第一

第十回

梶本裕之

日程

講義番号	講義日	講義内容	pdf	video	レポート締め切り
1	10/5	周期関数、フーリエ級数の定義	[pdf](2022年版)	video	10/12
2	10/12	フーリエ級数の計算例	[pdf](2022年版)	video	10/19
3	10/19	複素フーリエ級数、直交関数系	[pdf](2022年版)	video	10/26
4	10/26	周期関数のたたみこみ、パーセバルの等式	[pdf](2022年版)	video	11/2
5	11/2	非周期関数、フーリエ変換の定義・計算例	[pdf](2021年版)	video	11/9
6	11/9	フーリエ変換の性質	[pdf](2022年版)	video	11/16
7	11/16	デルタ関数とフーリエ変換、たたみこみのフーリエ変換、パーセバルの等式	[pdf](2022年版)	video	11/23
-	11/23	調布祭準備			
-	11/30	休講日（クオーター制との調整による）			
-	12/7	中間確認テストとその解説（前半。現在は大学を予定）	中間確認テスト用問題集	[pdf](2022年版)	
-	12/14	出張による休講			
8	12/21	離散時間信号と離散時間フーリエ変換（教科書外）	[pdf](2022年版)	video	1/4
9	1/4	離散フーリエ変換（教科書外）	[pdf](2022年版)	video	1/11
10	1/11	離散フーリエ変換の性質（教科書外）	[pdf](2022年版)	video	1/18
11	1/18	サンプリング定理	[pdf](2022年版)	video	1/25
12	1/25	ラプラス変換の定義と性質	[pdf](2022年版)	video	2/1
13	2/1	線形常微分方程式のラプラス変換による解法	[pdf](2022年版)	video	2/8
-	2/8	期末確認テストとその解説（後半。現在は大学を予定）	期末テスト用問題集	[pdf](2022年版)	

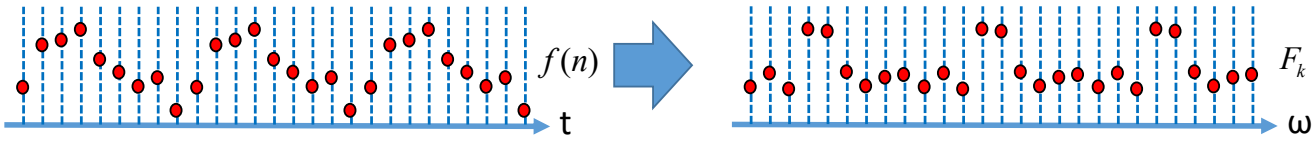
今日の目標

離散フーリエ変換の性質を知る

前回：離散フーリエ変換

$$F_k = \sum_{n=0}^{N-1} f(n)\exp(-j\frac{2\pi}{N}kn) \qquad f(n) = \frac{1}{N}\sum_{k=0}^{N-1} F_k \cdot \exp(j\frac{2\pi}{N}kn)$$

元の離散信号を周期的な信号とみなすことで
離散的かつ周期的な周波数信号に変換できる.



$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} W_4^{0\cdot0} & W_4^{0\cdot1} & W_4^{0\cdot2} & W_4^{0\cdot3} \\ W_4^{1\cdot0} & W_4^{1\cdot1} & W_4^{1\cdot2} & W_4^{1\cdot3} \\ W_4^{2\cdot0} & W_4^{2\cdot1} & W_4^{2\cdot2} & W_4^{2\cdot3} \\ W_4^{3\cdot0} & W_4^{3\cdot1} & W_4^{3\cdot2} & W_4^{3\cdot3} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \qquad \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \frac{1}{4}\begin{bmatrix} W_4^{-0\cdot0} & W_4^{-0\cdot1} & W_4^{-0\cdot2} & W_4^{-0\cdot3} \\ W_4^{-1\cdot0} & W_4^{-1\cdot1} & W_4^{-1\cdot2} & W_4^{-1\cdot3} \\ W_4^{-2\cdot0} & W_4^{-2\cdot1} & W_4^{-2\cdot2} & W_4^{-2\cdot3} \\ W_4^{-3\cdot0} & W_4^{-3\cdot1} & W_4^{-3\cdot2} & W_4^{-3\cdot3} \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

この離散フーリエ変換は，結局**正方行列によるデータ変換**であり，
逆離散フーリエ変換は**逆行列**となる.

（復習）元が4個のデータの場合

$$F_k = f(0)W_4^{k\cdot 0} + f(1)W_4^{k\cdot 1} + f(2)W_4^{k\cdot 2} + f(3)W_4^{k\cdot 3}$$

ただし $W_N = \exp(-j\frac{2\pi}{N})$

$W_4^{\,0} = 1$

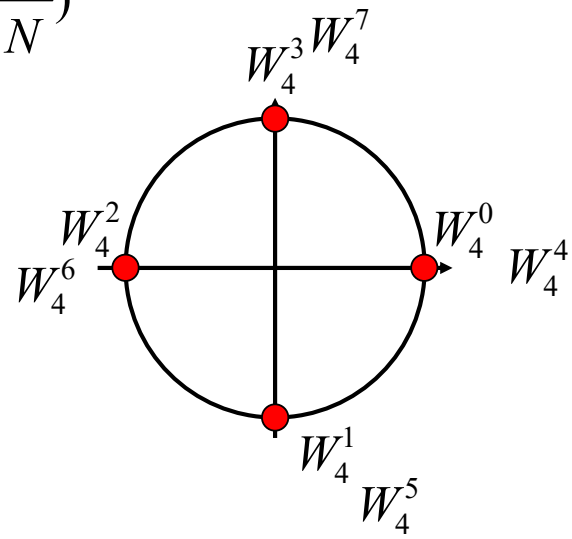
$W_4^{\,1} = \exp(-j\frac{2\pi}{4}) =$

$W_4^{\,2} = \exp(-j\frac{4\pi}{4}) =$

$W_4^{\,3} = \exp(-j\frac{6\pi}{4}) =$

$W_4^{\,4} = \exp(-j\frac{8\pi}{4}) =$

$\vdots$

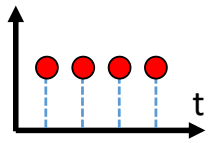


W_N^0 から W_N^N で複素単位円を一周する.

（復習）元が4個のデータの場合

元データが $\mathbf{f} = [1 \quad 1 \quad 1 \quad 1]^T$ だと

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

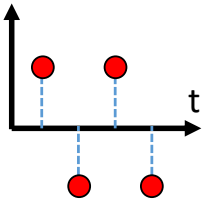


つまり F_0 だけが成分を持つ.

直流成分だけであることを意味している.

元データが $\mathbf{f} = [1 \quad -1 \quad 1 \quad -1]^T$ だと

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$



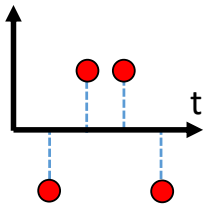
つまり F_2 だけが成分を持つ.

ある高い周波数成分だけであることを意味している.

（復習）元が4個のデータの場合

元データが $\mathbf{f}=\begin{bmatrix}-1 & 1 & 1 & -1\end{bmatrix}^T$ だと

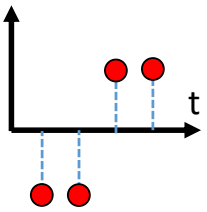
$$\begin{bmatrix}F_0\\F_1\\F_2\\F_3\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & 1\\1 & -j & -1 & j\\1 & -1 & 1 & -1\\1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}=\begin{bmatrix}\\\\\\\end{bmatrix}$$



F_1とF_3が同じ大きさを持っている（あとで振り返る）

元データが $\mathbf{f}=\begin{bmatrix}-1 & -1 & 1 & 1\end{bmatrix}^T$ だと

$$\begin{bmatrix}F_0\\F_1\\F_2\\F_3\end{bmatrix}=\begin{bmatrix}1 & 1 & 1 & 1\\1 & -j & -1 & j\\1 & -1 & 1 & -1\\1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}-1\\-1\\1\\1\end{bmatrix}=\begin{bmatrix}\\\\\\\end{bmatrix}$$



F_1とF_3が同じ大きさを持っている（あとで振り返る）

DFTの性質（１）周期性（復習）

$$F_k=\sum_{n=0}^{N-1}f(n)\exp(-j\frac{2\pi}{N}kn)$$

$$F_{k+N}=$$

$$=$$

$$=$$

$$=$$

DFTの性質（２）対称性（元データ実数時）

$$F_k = \sum_{n=0}^{N-1} f(n)\exp(-j\frac{2\pi}{N}kn)$$

元データf(n)が実数列である時,

$$\overline{F_{-k}} =$$

$$=$$

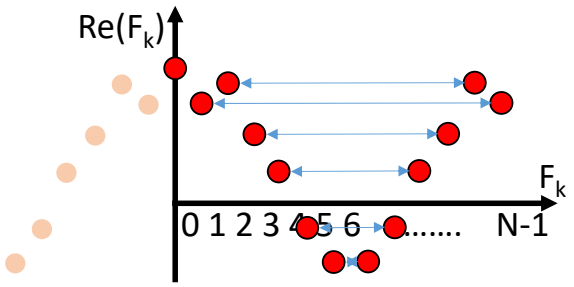
先の周期性 $F_{k+N} = F_k$ とあわせると

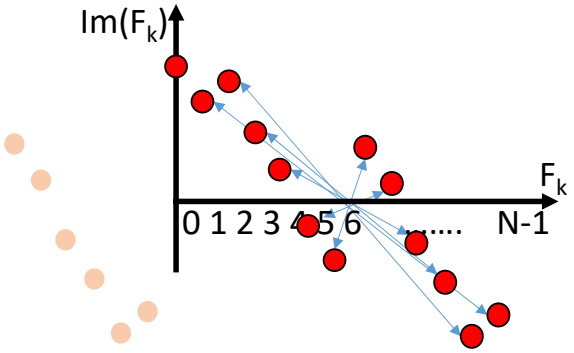
$$\overline{F_{N-k}} = F_k$$ あるいは,

DFTの性質（２）対称性（元データ実数時）続

元データf(n)が実数列である時,

$$\begin{cases} \mathrm{Re}(F_{N-k}) = \mathrm{Re}(F_k) & \text{左右対称となる．あるいは}\textbf{偶関数} \\ \mathrm{Im}(F_{N-k}) = -\mathrm{Im}(F_k) & \text{左右反対称となる．あるいは}\textbf{奇関数} \end{cases}$$





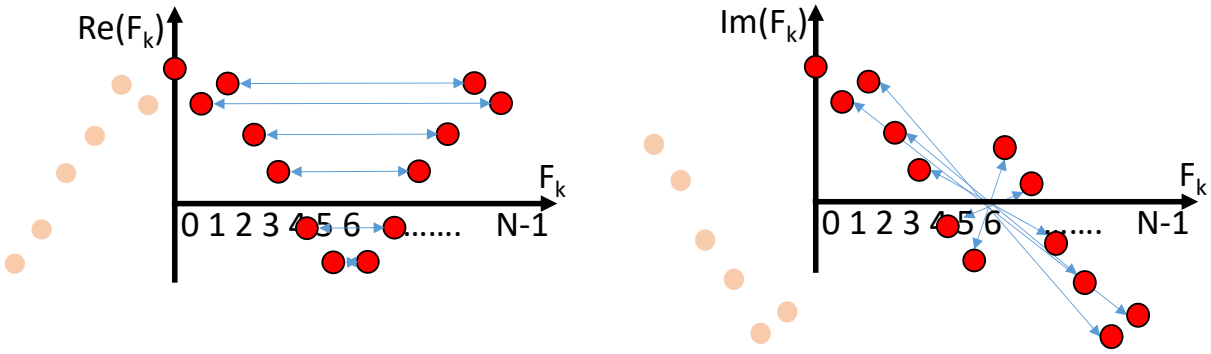
N=4の時の結果を見直すと…

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix} \qquad \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2+2j \\ 0 \\ -2-2j \end{bmatrix}$$

自由度に関する整合性

$$\begin{cases} \mathrm{Re}(F_{N-k}) = \mathrm{Re}(F_k) \\ \mathrm{Im}(F_{N-k}) = -\mathrm{Im}(F_k) \end{cases}$$

N個のうちN/2個しか独立ではない
ことを意味する



- 元のデータf(n): N個の独立なデータ．実数なので自由度N
- DFTの結果F_k: 実数部，虚数部共にN/2個の独立なデータ．

つまりDFTによって自由度に変化がないことを意味している．

DFTの性質（３）時間シフト

$$F_k = \sum_{n=0}^{N-1} f(n) \exp(-j\frac{2\pi}{N}kn)$$

$$\sum_{n=0}^{N-1} f(n-n_0) \exp(-j\frac{2\pi}{N}kn)$$

$=$

$=$

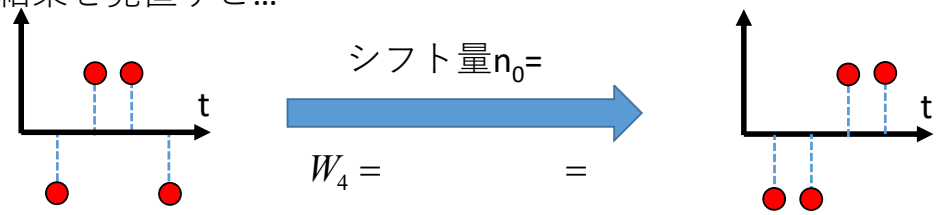
$=$ $=$ ただし $W_N = \exp(-j\frac{2\pi}{N})$

時間的にずれた信号→DFTは$\exp(-j\frac{2\pi}{N}kn_0)$の乗算

DFTの性質（３）時間シフト（例）

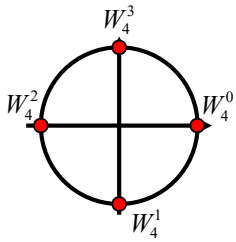
$$f(n-n_0) \Rightarrow F_k \exp(-j\frac{2\pi}{N}kn_0) = W_N^{kn_0} F_k \qquad \text{ただし } W_N = \exp(-j\frac{2\pi}{N})$$

N=4の時の結果を見直すと…



$W_4 =$ $=$

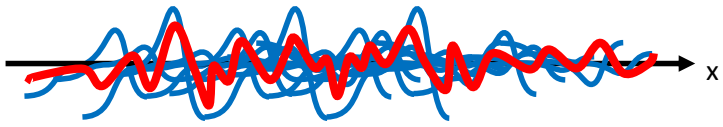
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix} \qquad \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2+2j \\ 0 \\ -2-2j \end{bmatrix}$$

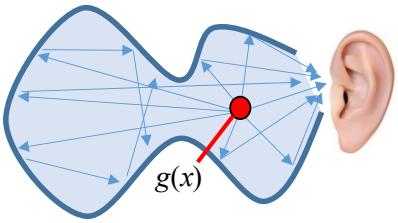


$$\begin{bmatrix} W_4^{0\cdot1} \\ W_4^{1\cdot1} \\ W_4^{2\cdot1} \\ W_4^{3\cdot1} \end{bmatrix} = \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix} \qquad \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix} \times \begin{bmatrix} 1 \\ -j \\ -1 \\ j \end{bmatrix} = \begin{bmatrix} 0 \\ -2+2j \\ 0 \\ -2-2j \end{bmatrix}$$

たしかに成立している

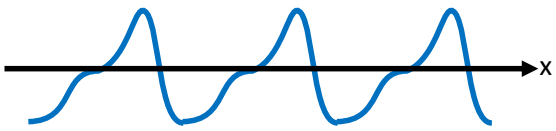
（復習）原音と響きの定式化



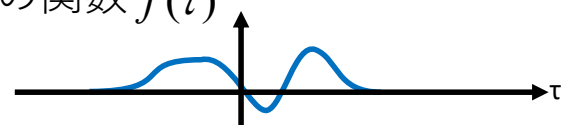


「遅れ時間τに対して振幅がどれだけか」の反響の関数があるなら，

原音 $g(x)$



反響の関数 $f(\tau)$



τだけ遅れた反響音成分は $f(\tau)g(x-\tau)$

遅れτに対応した反響の強さ　τだけ遅れた原信号

これらが**すべてのτで発生し，加算される**から，最終的な信号は

$$\int_{\tau=-\infty}^{\infty} f(\tau)g(x-\tau)d\tau$$

これを**関数f(x)とg(x)のたたみ込み積分**とよぶ

DFTの性質（４）たみ込み

$$\int_{\tau=-\infty}^{\infty} f(\tau)g(x-\tau)d\tau$$

$$H_k =$$

$$=$$

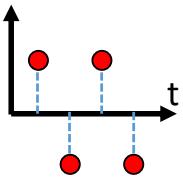
$$=$$

$$=$$ $$n'=n-i$$

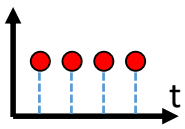
よって時間領域の畳込みは，周波数領域では単なる積となる
これまで繰り返し出てきた性質のDFT版

DFTの性質（４）たみ込み（例）

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$h(n) = \sum_{i=0}^{N-1} f(i)g(n-i)$$

$f(n) = [1 \quad -1 \quad 1 \quad -1]$ と $g(n) = [1 \quad 1 \quad 1 \quad 1]$ のたたみ込みは

$h(0) =$ $=$ $=$ $=$

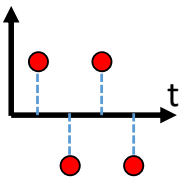
$h(1) =$ $=$ $=$ $=$

$h(2) =$ $=$ $=$ $=$

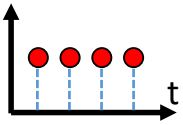
$h(3) =$ $=$ $=$

DFTの性質（４）たたみ込み（例：つづき）

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix}$$



$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$f(n)=[1 \quad -1 \quad 1 \quad -1]$ と $g(n)=[1 \quad 1 \quad 1 \quad 1]$ のたたみ込みは $h(n)=[0 \quad 0 \quad 0 \quad 0]$

よってその離散フーリエ変換は当然，$H_k=[0 \quad 0 \quad 0 \quad 0]$

一方で，$H_k=F_kG_k$ から $H_k=[\quad]*[\quad]=[\quad]$

よって，確かにf(n)とg(n)のたたみ込みの結果をDFTして得られた結果と，
FkとGkを直接掛け算して得られた結果は一致している．

DFTの性質（５）パーセバルの等式

$$\sum_{k=0}^{N-1}\overline{F_k}G_k =$$

$$=$$

ここで $\sum_{k=0}^{N-1}\exp(j\frac{2\pi}{N}k(m-n))=\begin{cases} N & m=n \\ 0 & m\neq n \end{cases}$ （この式は先週の逆DFTの導出時に導いた）

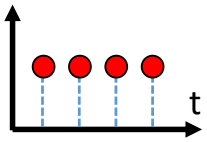
$$\therefore \sum_{k=0}^{N-1}\overline{F_k}G_k =$$

f(n)=g(n) の時，

これはDFTでエネルギーが保存されることを意味する

DFTの性質（５）パーセバルの等式（例）

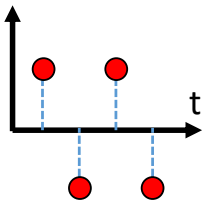
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$$\sum_{n=0}^{N-1} |f(n)|^2 =$$

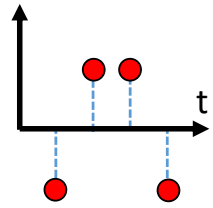
$$\sum_{k=0}^{N-1} |F_k|^2 =$$

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix}$$



$$\sum_{k=0}^{N-1} |F_k|^2 =$$

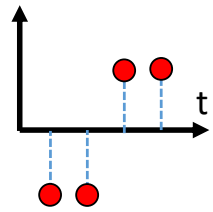
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix}$$



$$\sum_{k=0}^{N-1} |F_k|^2 =$$

$$=$$

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2+2j \\ 0 \\ -2-2j \end{bmatrix}$$



$$\sum_{k=0}^{N-1} |F_k|^2 =$$

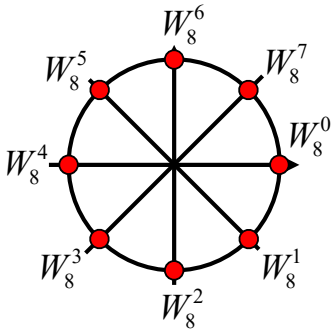
$$=$$

DFTの計算量

元データが8個の場合

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \begin{bmatrix} W_8^{0\cdot0} & W_8^{0\cdot1} & W_8^{0\cdot2} & W_8^{0\cdot3} & W_8^{0\cdot4} & W_8^{0\cdot5} & W_8^{0\cdot6} & W_8^{0\cdot7} \\ W_8^{1\cdot0} & W_8^{1\cdot1} & W_8^{1\cdot2} & W_8^{1\cdot3} & W_8^{1\cdot4} & W_8^{1\cdot5} & W_8^{1\cdot6} & W_8^{1\cdot7} \\ W_8^{2\cdot0} & W_8^{2\cdot1} & W_8^{2\cdot2} & W_8^{2\cdot3} & W_8^{2\cdot4} & W_8^{2\cdot5} & W_8^{2\cdot6} & W_8^{2\cdot7} \\ W_8^{3\cdot0} & W_8^{3\cdot1} & W_8^{3\cdot2} & W_8^{3\cdot3} & W_8^{3\cdot4} & W_8^{3\cdot5} & W_8^{3\cdot6} & W_8^{3\cdot7} \\ W_8^{4\cdot0} & W_8^{4\cdot1} & W_8^{4\cdot2} & W_8^{4\cdot3} & W_8^{4\cdot4} & W_8^{4\cdot5} & W_8^{4\cdot6} & W_8^{4\cdot7} \\ W_8^{5\cdot0} & W_8^{5\cdot1} & W_8^{5\cdot2} & W_8^{5\cdot3} & W_8^{5\cdot4} & W_8^{5\cdot5} & W_8^{5\cdot6} & W_8^{5\cdot7} \\ W_8^{6\cdot0} & W_8^{6\cdot1} & W_8^{6\cdot2} & W_8^{6\cdot3} & W_8^{6\cdot4} & W_8^{6\cdot5} & W_8^{6\cdot6} & W_8^{6\cdot7} \\ W_8^{7\cdot0} & W_8^{7\cdot1} & W_8^{7\cdot2} & W_8^{7\cdot3} & W_8^{7\cdot4} & W_8^{7\cdot5} & W_8^{7\cdot6} & W_8^{7\cdot7} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

$$W_N = \exp(-j\frac{2\pi}{N})$$



8x8行列と8x1ベクトルの演算なので，8x8=64回の乗算が発生．

40kHzでサンプリングされた1秒のデータ：40000個．

回の乗算

3GHzのCPUが1クロックで1回乗算できるとすると，

1600000000/3000000000≒0.5秒かかる．

10秒分のデータだと？？

DFTの計算量を削減する

$$F_k = \sum_{n=0}^{N-1} f(n)\exp(-j\frac{2\pi}{N}kn) = \sum_{n=0}^{N-1} f(n)W_N^{kn} \qquad W_N = \exp(-j\frac{2\pi}{N})$$

Nが偶数である場合を考える。k=2m、すなわち偶数番目の項は、

$$F_{2m} = \sum_{n=0}^{\frac{N}{2}-1} f(n)W_N^{2mn} + \sum_{n=\frac{N}{2}}^{N-1} f(n)W_N^{2mn}$$

前半と後半に分割

$$=$$

n=n'+N/2と変数変換したうえでn'をnに書き戻す

$$=$$

∵W_Nの周期性から

$$=$$

$$\because W_N^{2mn} = \exp\left(-j\frac{2\pi}{N}2mn\right) = \exp\left(-j\frac{2\pi}{N/2}mn\right)$$

これは新たなN/2個の信号$f(n)+f(n+\frac{N}{2})$のフーリエ変換である。

DFTの計算量を削減する

$$F_k = \sum_{n=0}^{N-1} f(n)\exp(-j\frac{2\pi}{N}kn) = \sum_{n=0}^{N-1} f(n)W_N^{kn} \qquad W_N = \exp(-j\frac{2\pi}{N})$$

k=2m+1、すなわち奇数番目の項は、

$$F_{2m+1} = \sum_{n=0}^{\frac{N}{2}-1} f(n)W_N^{(2m+1)n} + \sum_{n=\frac{N}{2}}^{N-1} f(n)W_N^{(2m+1)n}$$

$$=$$

$$=$$

$$=$$

∵ $W_N^{N/2}$=-1から

$$=$$

$$\because W_N^{2mn} = \exp\left(-j\frac{2\pi}{N}2mn\right) = \exp\left(-j\frac{2\pi}{N/2}mn\right)$$

これは新たなN/2個の信号$f(n)-f(n+\frac{N}{2})$のフーリエ変換にW_N^nをかけたもの。

DFTの計算量を削減する

$$F_{2m} = \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) + f(n+\frac{N}{2}) \right) W_N^{2mn}$$ N/2個の信号$f(n) + f(n+\frac{N}{2})$のフーリエ変換

$$F_{2m+1} = W_N^n \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) - f(n+\frac{N}{2}) \right) W_N^{(2m+1)n}$$ N/2個の信号$f(n) - f(n+\frac{N}{2})$のフーリエ変換（に W_N^n をかけたもの）

つまり、**N個**の信号のフーリエ変換（乗算がN^2個必要）を、
２つの、**N/2**個の信号のフーリエ変換（乗算が$N^2/2$個必要）に変換できた。

さらにN/2個の信号を、N/4個ずつに分割できれば。。。
さらにさらにN/4個の信号を、N/8個ずつに分割できれば。。。

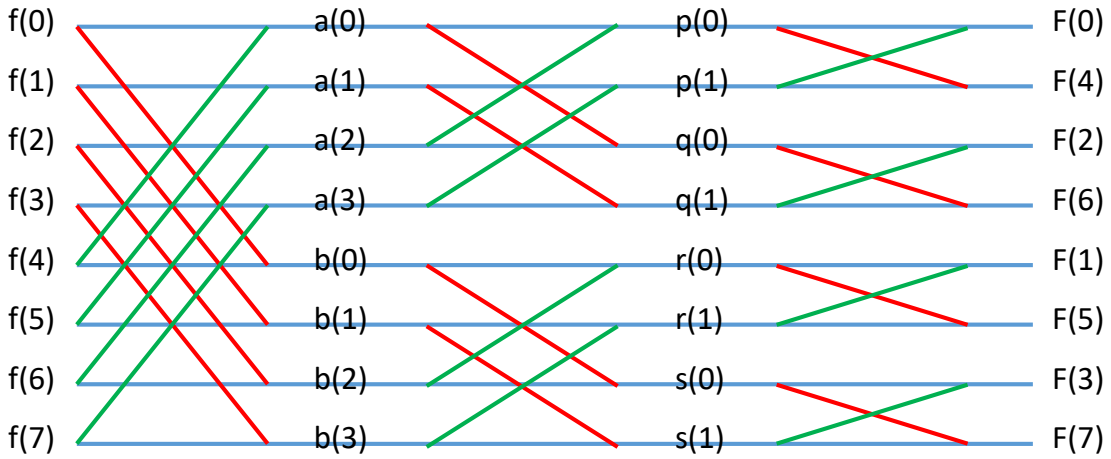
このような1/2分割が最後までできるのは、データの個数が**2の階乗**である場合。

このように、1/2分割による計算量の削減を行う手法を
高速フーリエ変換、FFT(Fast Fourier Tranform)と呼ぶ。

N=8の場合

$$F_{2m} = \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) + f(n+\frac{N}{2}) \right) W_N^{2mn}$$ N/2個の信号$f(n) + f(n+\frac{N}{2})$のフーリエ変換

$$F_{2m+1} = W_N^n \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) - f(n+\frac{N}{2}) \right) W_N^{2mn}$$ N/2個の信号$f(n) - f(n+\frac{N}{2})$のフーリエ変換

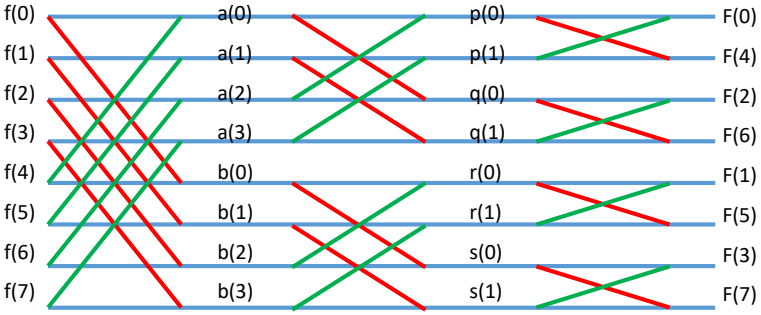


これをフーリエ変換するには

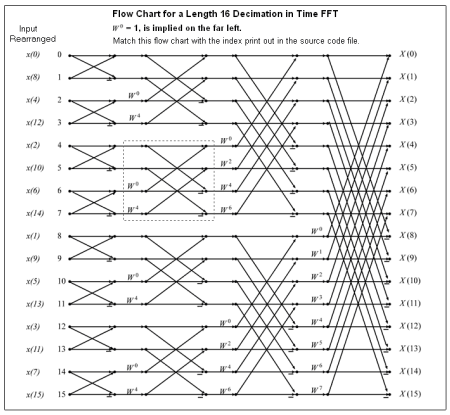
a(0)=f(0)+f(4), …
b(0)=f(0)-f(4), …
をフーリエ変換すれば
よく、そのためには

p(0)=a(0)+a(2), …q(0)=a(0)-a(2) ,…
r(0)=b(0)+b(2), …s(0)=b(0)-b(2) ,…
をフーリエ変換すればよい。

バタフライ演算とFFT



書き方の流儀は色々



16点の例 Burrus, C. Fast Fourier Transforms, Connexions Web site. http://cnx.org/content/col10550/1.22/

http://iowahills.com/Example%20Code/FFTButterfly.html

$O(n^2)$ の乗算回数が $O(n\log(n))$ に削減される（加減算も削減）

例えば1024点のデータの場合，約100万回→約5000回に削減(1/200)
例えば65536点のデータの場合，約40億回→約52万回に削減(1/10000)

バタフライ演算をするため，元データの個数は2^N個である必要．
現実的には0で埋めて数を揃える．

今日のまとめ

- DFTの性質をみた．これまでのフーリエ◯◯と同様の性質を確認した．
- DFTの計算量に関して見積もり，バタフライ演算によって劇的に低減されることを確認した（FFT).

次回は離散フーリエ変換と信号処理

今日のレポート

(畳み込みの性質）

２つのベクトル[-1, 1,1,-1]と[-1,-1,1, 1]の畳み込みを求めて離散フーリエ変換を行い、その結果が確かに、それぞれのベクトルの離散フーリエ変換の掛け算になっていることを示せ

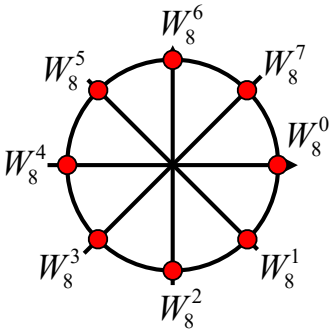
レポートは紙に書いたものを写真にとり、指定のフォームからアップロードする。Googleアカウントによりアップロードするので、大学が提供しているアカウントを使用すること。画像ファイルは5MB以下に抑えること。

提出締め切り：講義日から一週間以内

DFTの計算量を削減する

$$W_N = \exp(-j\frac{2\pi}{N})$$

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$



回転因子W_Nの規則性から $W_8^n = W_8^{n+8} = W_8^{n+16} = \cdots = W_8^{n \bmod 8}$ だから

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

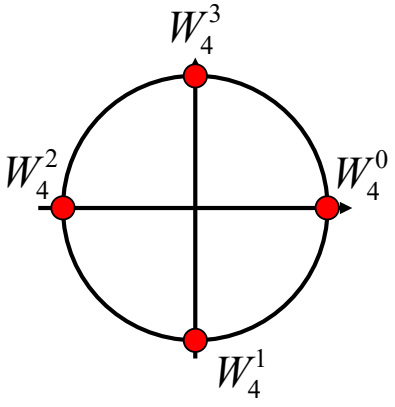
この規則性から，計算量を減らす工夫を見出す．

DFTの計算量を削減する：４点の場合

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} W_4^{0\cdot0} & W_4^{0\cdot1} & W_4^{0\cdot2} & W_4^{0\cdot3} \\ W_4^{1\cdot0} & W_4^{1\cdot1} & W_4^{1\cdot2} & W_4^{1\cdot3} \\ W_4^{2\cdot0} & W_4^{2\cdot1} & W_4^{2\cdot2} & W_4^{2\cdot3} \\ W_4^{3\cdot0} & W_4^{3\cdot1} & W_4^{3\cdot2} & W_4^{3\cdot3} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix}$$

２列目と３列目を入れ替える

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & j \\ 1 & -1 \\ 1 & -j \end{bmatrix} \begin{bmatrix} f(0) \\ f(2) \\ f(1) \\ f(3) \end{bmatrix}$$

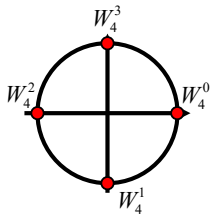


DFTの計算量を削減する

2つに分けて表記

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ f(2) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ -j & j \\ -1 & -1 \\ j & -j \end{bmatrix} \begin{bmatrix} f(1) \\ f(3) \end{bmatrix}$$

回転因子の性質を用いて



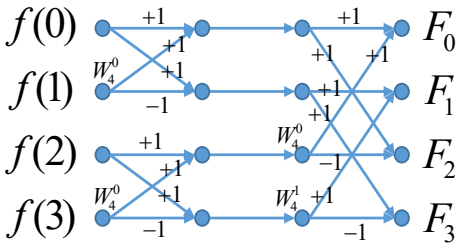
$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ W_4^0 \cdot f(2) \end{bmatrix} + \begin{bmatrix} & \\ & \\ & \\ & \end{bmatrix} \begin{bmatrix} f(1) \\ W_4^0 \cdot f(3) \end{bmatrix}$$

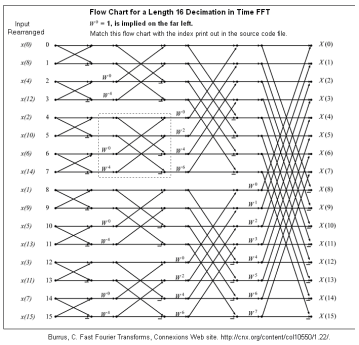
この式には, $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ という配列が4つ含まれることに注目

DFTの計算量を削減する

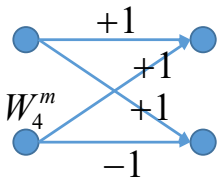
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ W_4^0 \cdot f(2) \end{bmatrix} + \begin{bmatrix} W_4^0 \cdot 1 & W_4^0 \cdot 1 \\ W_4^1 \cdot 1 & W_4^1 \cdot -1 \\ -W_4^0 \cdot 1 & -W_4^0 \cdot 1 \\ -W_4^1 \cdot 1 & -W_4^1 \cdot -1 \end{bmatrix} \begin{bmatrix} f(1) \\ W_4^0 \cdot f(3) \end{bmatrix}$$

バタフライ演算とFFT





16点の場合
http://iowahills.com/Example%20Code/FFTButterfly.html



バタフライユニット：
乗算回数は１回

４つのバタフライユニットだから，乗算回数は4回
→もともと4x4=16回だったから1/4に.

バタフライ演算による高速化：FFT(Fast Fourier Transform)と呼ぶ.

$O(n^2)$ の乗算回数が $O(n\log(n))$ に削減される（加減算も削減される）

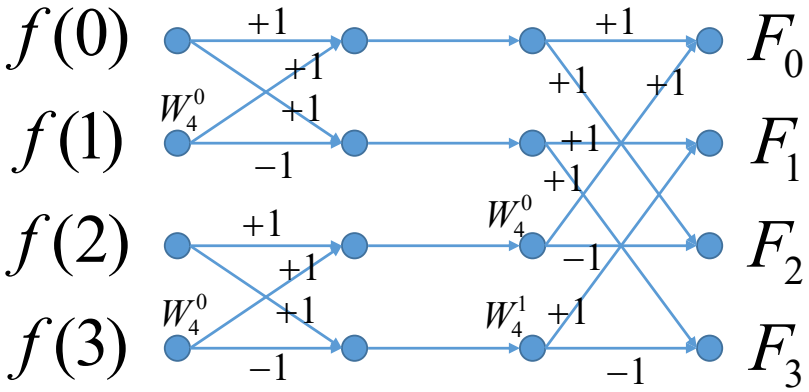
例えば1024点のデータの場合，約100万回→約5000回に削減(1/200)
例えば65536点のデータの場合，約40億回→約52万回に削減(1/10000)

バタフライ演算をするため，元データの個数は2^N個である必要.
現実的には0で埋めて数を揃える.

DFTの計算量を削減する

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} f(0) \\ W_4^0 \cdot f(2) \end{bmatrix} + \begin{bmatrix} W_4^0 \cdot 1 & W_4^0 \cdot 1 \\ W_4^1 \cdot 1 & W_4^1 \cdot -1 \\ -W_4^0 \cdot 1 & -W_4^0 \cdot 1 \\ -W_4^1 \cdot 1 & -W_4^1 \cdot -1 \end{bmatrix} \begin{bmatrix} f(1) \\ W_4^0 \cdot f(3) \end{bmatrix}$$

この演算は次のように表せる



バタフライ演算と呼ぶ