



# 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
1	10/1	周期関数、フーリエ級数の定義
2	10/8	フーリエ級数の計算例
3	10/15	複素フーリエ級数、直交関数系
-	10/22	体育祭
4	10/29	周期関数のたたみこみ、パーセバルの等式
5	11/5	非周期関数、フーリエ変換の定義・計算例
6	11/12	フーリエ変換の性質
7	11/19	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
-	11/26	中間確認問題（自習）
8	12/3	離散時間信号と離散時間フーリエ変換（教科書外）
9	12/10	離散フーリエ変換（教科書外）
10	12/17	離散フーリエ変換の性質（教科書外）
11	1/7	サンプリング定理
12	1/14	ラプラス変換の定義と性質
13	1/21	線形常微分方程式のラプラス変換による解法
-	1/28	期末テスト準備（自習）
-	2/4	期末確認テスト（全範囲。現在は大学を予定）



# 今日の目標

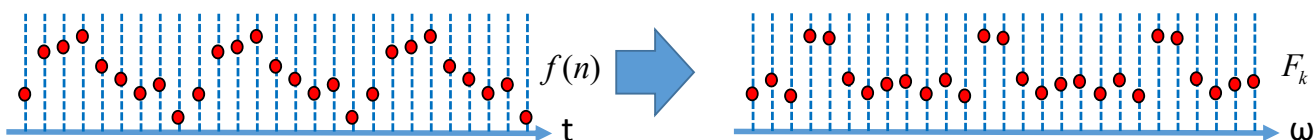
## 離散フーリエ変換の性質を知る



前回：離散フーリエ変換

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

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。



$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} W_4^{0 \cdot 0} & W_4^{0 \cdot 1} & W_4^{0 \cdot 2} & W_4^{0 \cdot 3} \\ W_4^{1 \cdot 0} & W_4^{1 \cdot 1} & W_4^{1 \cdot 2} & W_4^{1 \cdot 3} \\ W_4^{2 \cdot 0} & W_4^{2 \cdot 1} & W_4^{2 \cdot 2} & W_4^{2 \cdot 3} \\ W_4^{3 \cdot 0} & W_4^{3 \cdot 1} & W_4^{3 \cdot 2} & W_4^{3 \cdot 3} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \quad \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} W_4^{-0 \cdot 0} & W_4^{-0 \cdot 1} & W_4^{-0 \cdot 2} & W_4^{-0 \cdot 3} \\ W_4^{-1 \cdot 0} & W_4^{-1 \cdot 1} & W_4^{-1 \cdot 2} & W_4^{-1 \cdot 3} \\ W_4^{-2 \cdot 0} & W_4^{-2 \cdot 1} & W_4^{-2 \cdot 2} & W_4^{-2 \cdot 3} \\ W_4^{-3 \cdot 0} & W_4^{-3 \cdot 1} & W_4^{-3 \cdot 2} & W_4^{-3 \cdot 3} \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

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

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

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

$$W_4^0 = 1$$

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

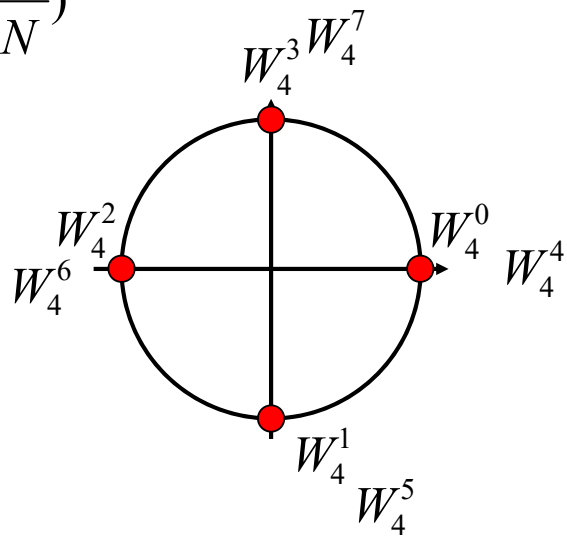
$$W_4^2 = \exp(-j \frac{4\pi}{4}) =$$

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

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

⋮

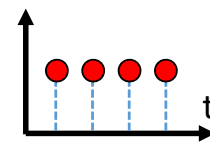
$W_N^0$  から  $W_N^N$  で複素単位円を一周する.



(復習) 元が4個のデータの場合

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

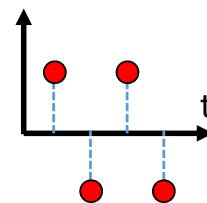
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$



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  $\mathbf{f} = [1 \ -1 \ 1 \ -1]^T$  だと

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



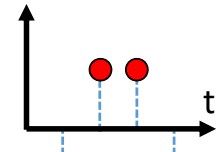
つまり  $F_2$  だけが成分を持つ。  
**ある高い周波数成分**だけであることを意味している。



(復習) 元が4個のデータの場合

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

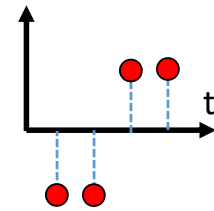
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} \dots \\ \dots \\ \dots \\ \dots \end{bmatrix}$$



$F_1$ 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  $\mathbf{f} = [-1 \ -1 \ 1 \ 1]^T$  だと

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



$F_1$ 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

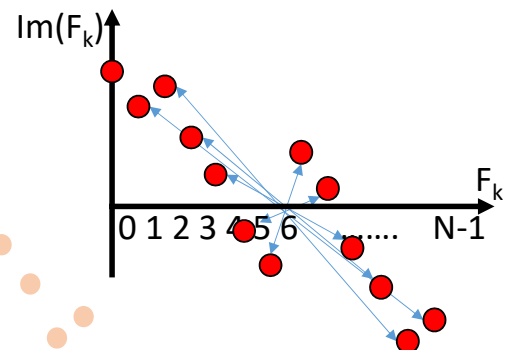
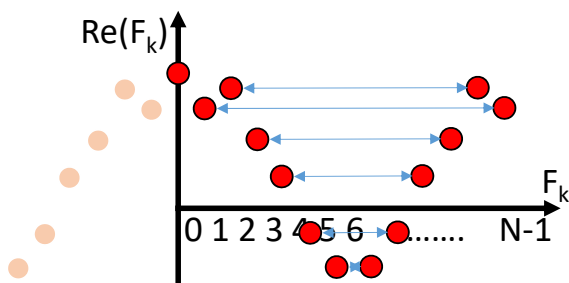
$$\overline{F_{N-k}} = F_k \quad \text{あるいは,} \quad \left\{ \right.$$



## 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,

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



$N=4$  の時の結果を見直すと...

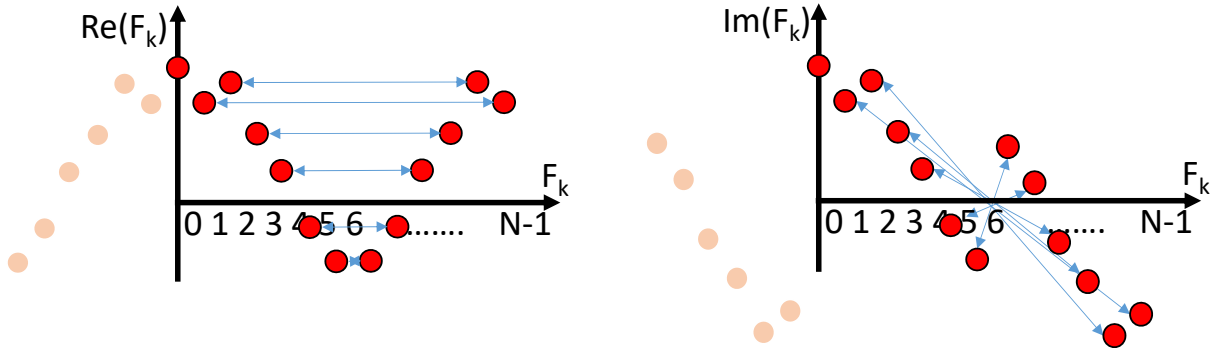
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix}$$

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



# 自由度に関する整合性

$$\begin{cases} \text{Re}(F_{N-k}) = \text{Re}(F_k) \\ \text{Im}(F_{N-k}) = -\text{Im}(F_k) \end{cases} \Rightarrow \text{N個のうちN/2個しか独立ではないことを意味する}$$



- 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

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

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

=

=

=

=

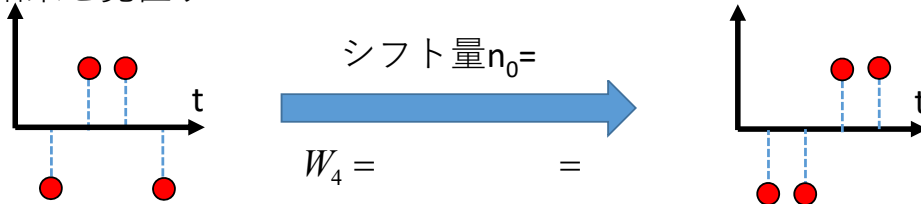
ただし  $W_N = \exp(-j \frac{2\pi}{N})$

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)

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

N=4の時の結果を見直すと...



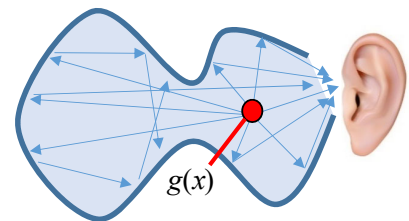
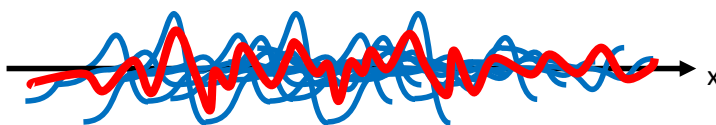
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix} \quad \begin{bmatrix} F_0 \\ F_1 \\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

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

たしかに成立している

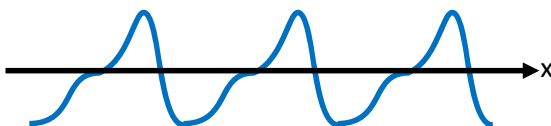


## (復習) 原音と響きの定式化

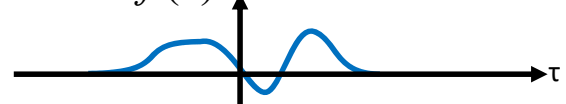


「遅れ時間 $\tau$ に対して振幅がどれだけか」の反響の関数があるなら、

原音  $g(x)$



反響の関数  $f(\tau)$



$\tau$ だけ遅れた反響音成分は  $f(\tau)g(x-\tau)$

遅れ $\tau$ に対応した反響の強さ  $\tau$ だけ遅れた原信号

これらがすべての $\tau$ 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）たたみ込み

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

$$H_k =$$

$$=$$

$$=$$

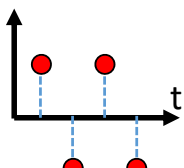
$$=$$

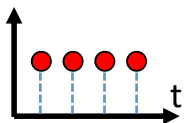
$$n' = n - i$$

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）

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


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


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

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

$$h(0) = \quad = \quad = \quad =$$

$$h(1) = \quad = \quad = \quad =$$

$$h(2) = \quad = \quad = \quad =$$

$$h(3) = \quad = \quad = \quad =$$





## DFTの性質（４）たたみ込み（例：つづき）

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

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

$f(n)=[1 \ -1 \ 1 \ -1]$ と $g(n)=[1 \ 1 \ 1 \ 1]$ のたたみ込みは $h(n)=[0 \ 0 \ 0 \ 0]$   
 よってその離散フーリエ変換は当然、 $H_k=[0 \ 0 \ 0 \ 0]$

一方で、 $H_k = F_k G_k$  から  $H_k = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} * \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$

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、 $F_k$ と $G_k$ 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

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

=

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

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

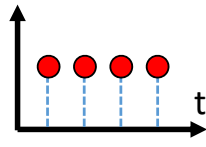
$f(n)=g(n)$  の時、

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)

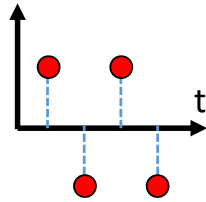
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



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

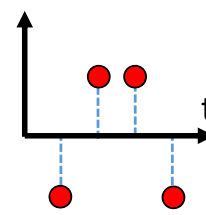
$$\sum_{k=0}^{N-1} |F_k|^2 =$$

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



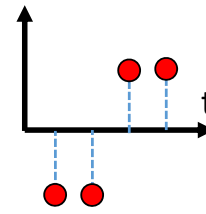
$$\sum_{k=0}^{N-1} |F_k|^2 =$$

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



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

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



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

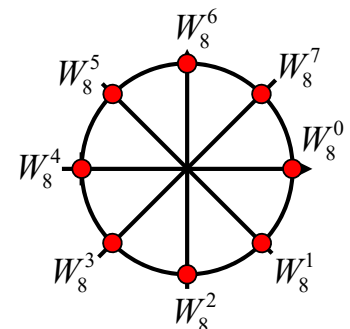


## DFTの計算量

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

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

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、

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

=

=

∵  $W_N$ の周期性から

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

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

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

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

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

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

つまり、**N個**の信号のフーリエ変換（乗算が**N<sup>2</sup>**個必要）を、  
2つの、**N/2個**の信号のフーリエ変換（乗算が**N<sup>2</sup>/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。

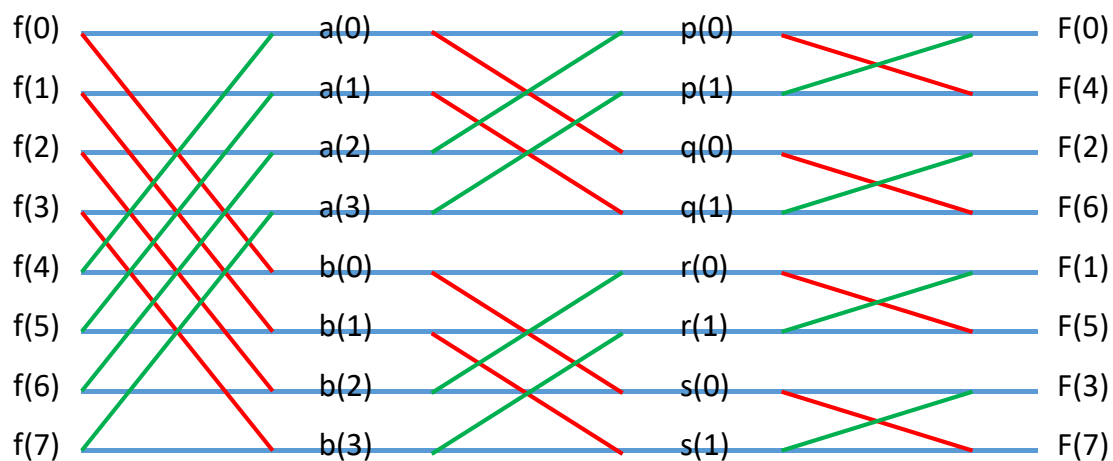
このように、**1/2**分割による計算量の削減を行う手法を  
高速フーリエ変換、**FFT(Fast Fourier Transform)**と呼ぶ。



## N=8の場合

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

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



これをフーリエ  
変換するには



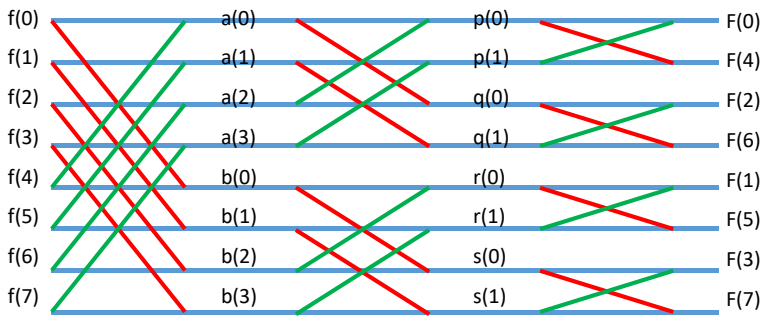
$a(0)=f(0)+f(4), \dots$   
 $b(0)=f(0)-f(4), \dots$   
をフーリエ変換すれば  
よく、そのためには



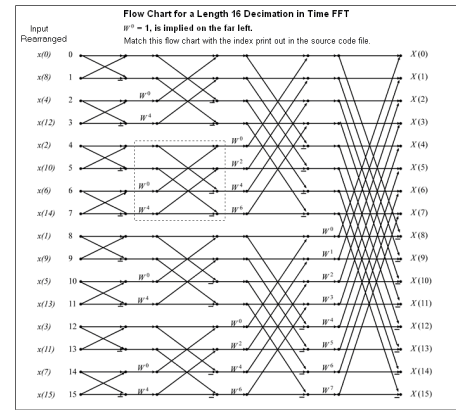
$p(0)=a(0)+a(2), \dots, q(0)=a(0)-a(2), \dots$   
 $r(0)=b(0)+b(2), \dots, s(0)=b(0)-b(2), \dots$   
をフーリエ変換すればよい。



# バタフライ演算とFFT



書き方の流儀は色々



16点の例

Burns, C. Fast Fourier Transforms, Conexions Web site: <http://cnx.org/content/col10650/1.22/>

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



バタフライユニット：乗算回数は1回

全体の乗算回数はバタフライユニットの個数となる。

$O(n^2)$  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、下記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。

<https://forms.gle/tovLwBV1eauiuoAVA>

提出締め切り：講義日から一週間以内

