



日程

講義番号	講義日	講義内容	pdf	video	レポート締め切り
1	10/3	周期関数、フーリエ級数の定義	[ pdf](2023年版)	video 	10/10
2	10/10	フーリエ級数の計算例	[ pdf](2023年版)	video 	10/17
3	10/17	複素フーリエ級数、直交関数系	[ pdf](2023年版)	video 	10/24
4	10/24	周期関数のたたみこみ、パーセバルの等式	[ pdf](2023年版)	video 	10/31
5	10/31 (オンデマンド)	非周期関数、フーリエ変換の定義・計算例	[ pdf](2023年版)	video 	11/7
6	11/7	フーリエ変換の性質	[ pdf](2023年版)	video 	11/14
7	11/14	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](2023年版)	video 	11/21
-	11/21	調布祭準備			
-	11/28	中間確認テストとその解説 (前半)	中間確認テスト用問題集	[ pdf](2022年版)	
-	12/5	休講日 (クォーター制との調整による)			
8	12/12	離散時間信号と離散時間フーリエ変換 (教科書外)	[ pdf](2023年版)	video 	12/19
9	12/19 (オンデマンド)	離散フーリエ変換 (教科書外)	[ pdf](2023年版)	video 	12/26
10	12/26	離散フーリエ変換の性質 (教科書外)	[ pdf](2023年版)	video 	1/9
11	1/9	サンプリング定理	[ pdf](2023年版)	video 	1/16
12	1/16	ラプラス変換の定義と性質	[ pdf](2023年版)	video 	1/23
13	1/23	線形常微分方程式のラプラス変換による解法	[ pdf](2023年版)	video 	1/30
-	1/30	期末確認テストとその解説 (後半)	期末テスト用問題集	[ pdf](2022年版)	

今日の目標

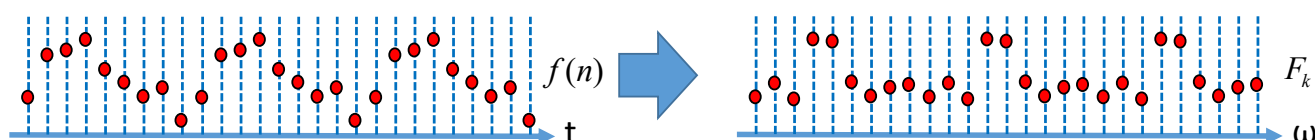
離散フーリエ変換の性質を知る



前回：離散フーリエ変換

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

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。



$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} W_4^{0.0} & W_4^{0.1} & W_4^{0.2} & W_4^{0.3} \\ W_4^{1.0} & W_4^{1.1} & W_4^{1.2} & W_4^{1.3} \\ W_4^{2.0} & W_4^{2.1} & W_4^{2.2} & W_4^{2.3} \\ W_4^{3.0} & W_4^{3.1} & W_4^{3.2} & W_4^{3.3} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} \quad \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} W_4^{-0.0} & W_4^{-0.1} & W_4^{-0.2} & W_4^{-0.3} \\ W_4^{-1.0} & W_4^{-1.1} & W_4^{-1.2} & W_4^{-1.3} \\ W_4^{-2.0} & W_4^{-2.1} & W_4^{-2.2} & W_4^{-2.3} \\ W_4^{-3.0} & W_4^{-3.1} & W_4^{-3.2} & W_4^{-3.3} \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

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

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

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

$$W_4^0 = 1$$

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

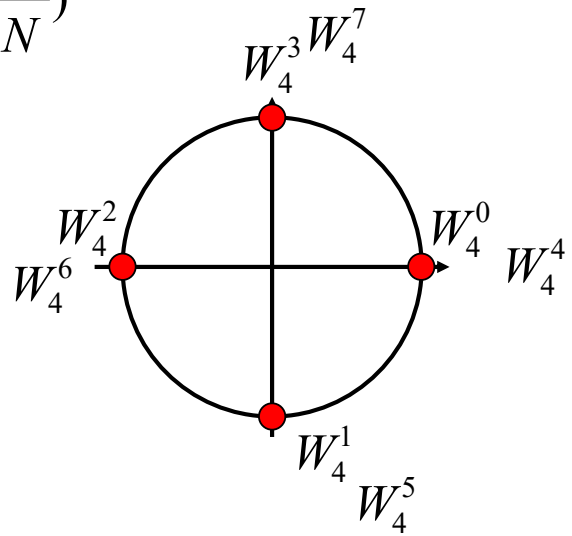
$$W_4^2 = \exp(-j \frac{4\pi}{4}) =$$

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

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

⋮

W_N^0 から W_N^N で複素単位円を一周する.



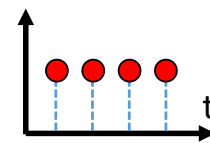
(復習) 元が4個のデータの場合

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

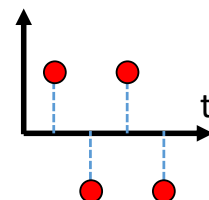
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

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.



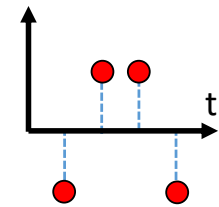
つまり F_2 だけが成分を持つ.
ある高い周波数成分だけであることを意味している.



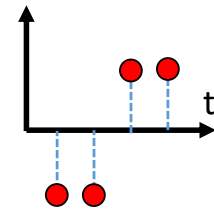
(復習) 元が4個のデータの場合

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

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



F_1 と F_3 が同じ大きさを持っている (あとで振り返る)



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

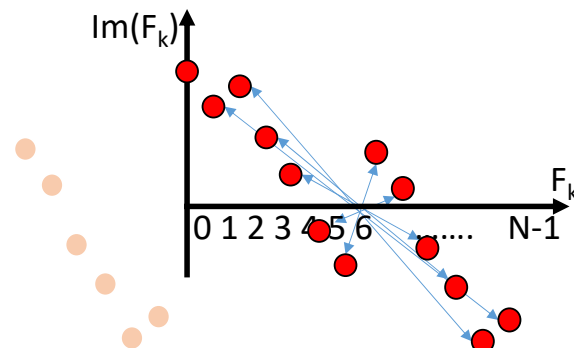
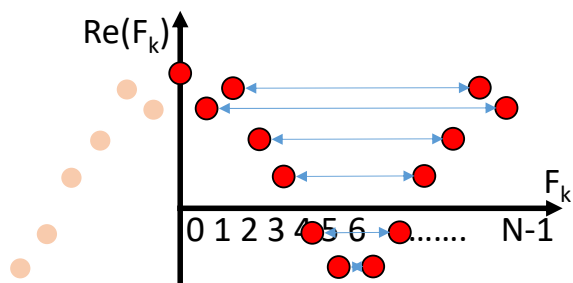
$$\overline{F_{N-k}} = F_k \quad \text{あるいは,} \quad \left\{ \begin{array}{l} \text{Re}(F_{N-k}) = \text{Re}(F_k) \\ \text{Im}(F_{N-k}) = -\text{Im}(F_k) \end{array} \right.$$



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,

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



N=4の時の結果を見直すと...

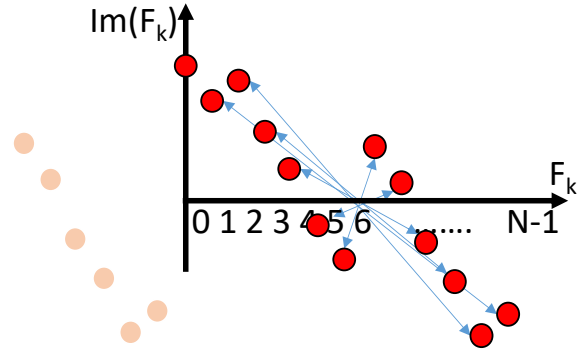
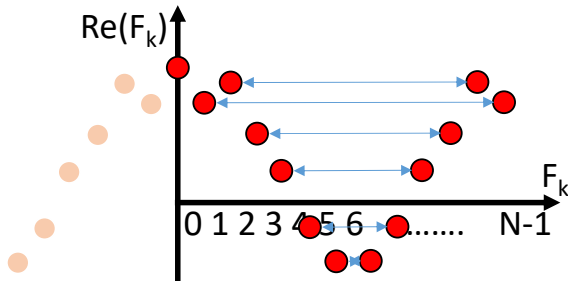
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix}$$

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



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

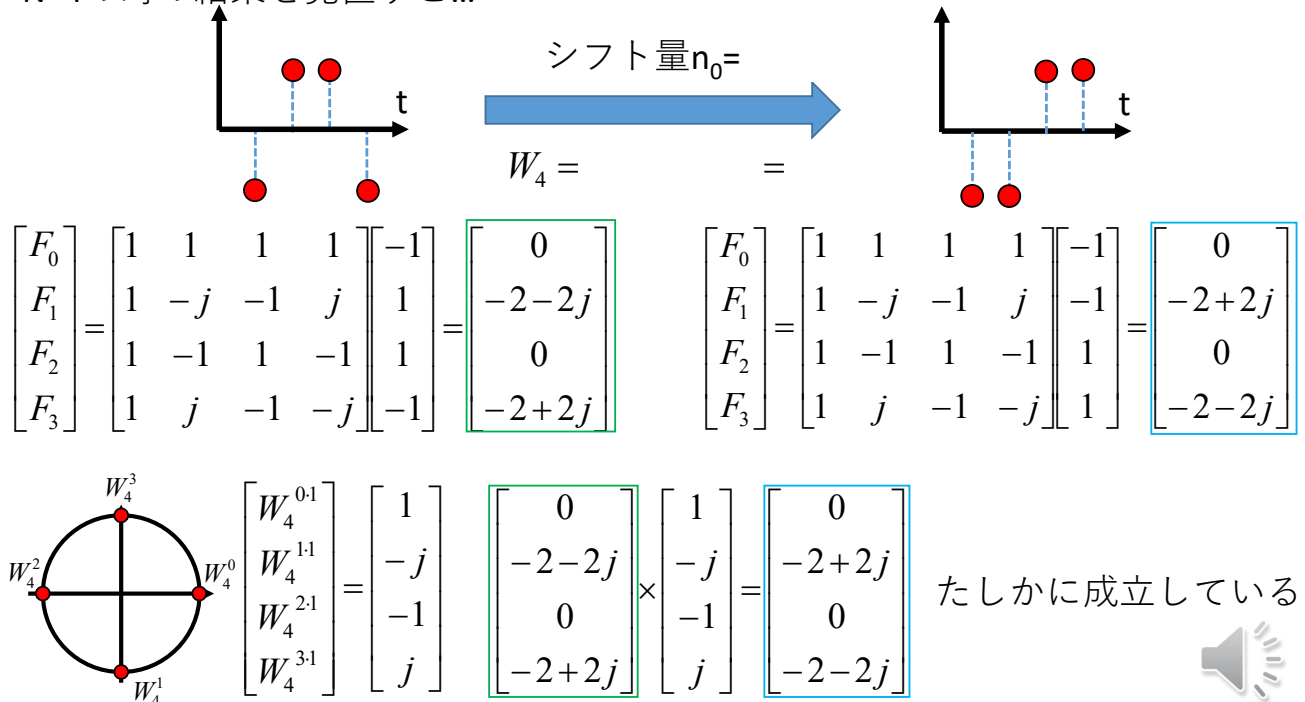
時間的にずれた信号 → DFTは $\exp(-j \frac{2\pi}{N} kn_0)$ の乗算



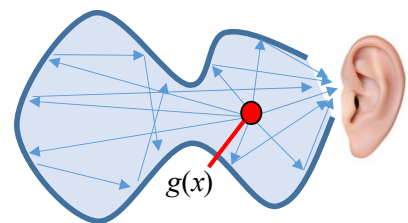
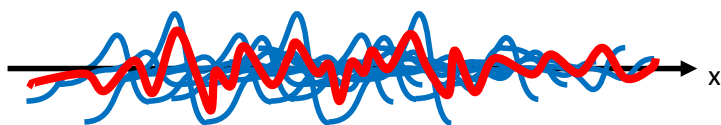
DFTの性質 (3) 時間シフト (例)

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

N=4の時の結果を見直すと...

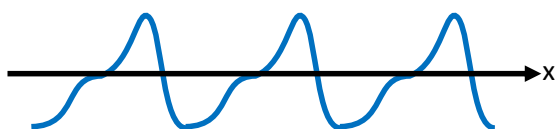


(復習) 原音と響きの定式化

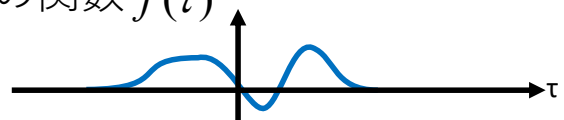


「遅れ時間 τ に対して振幅がどれだけか」の反響の関数があるなら、

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

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

$$H_k =$$

$$=$$

$$=$$

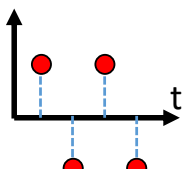
$$=$$

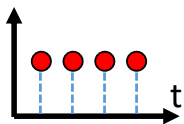
$$n'=n-i$$

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）

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


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


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

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

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

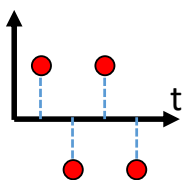
$$h(1) = \quad = \quad = \quad =$$

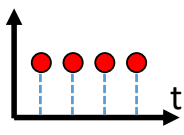
$$h(2) = \quad = \quad = \quad =$$

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



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 $H_k = \begin{bmatrix} \vdots \end{bmatrix} * \begin{bmatrix} \vdots \end{bmatrix} = \begin{bmatrix} \vdots \end{bmatrix}$

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

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

=

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)

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

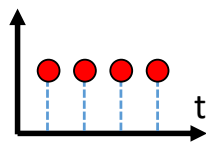
$f(n)=g(n)$ の時,

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)

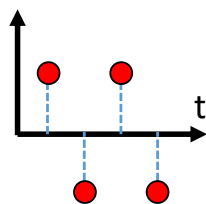
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



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

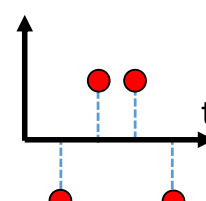
$$\sum_{k=0}^{N-1} |F_k|^2 =$$

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



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

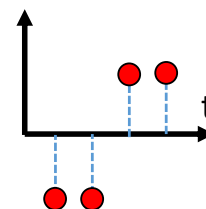
$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -2-2j \\ 0 \\ -2+2j \end{bmatrix}$$



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

=

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



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

=

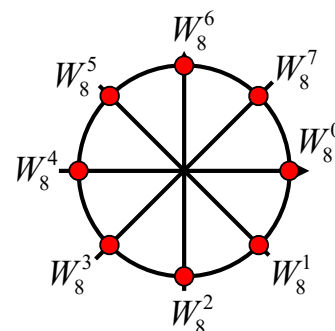


DFTの計算量

元データが8個の場合

$$\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \begin{bmatrix} W_8^{0.0} & W_8^{0.1} & W_8^{0.2} & W_8^{0.3} & W_8^{0.4} & W_8^{0.5} & W_8^{0.6} & W_8^{0.7} \\ W_8^{1.0} & W_8^{1.1} & W_8^{1.2} & W_8^{1.3} & W_8^{1.4} & W_8^{1.5} & W_8^{1.6} & W_8^{1.7} \\ W_8^{2.0} & W_8^{2.1} & W_8^{2.2} & W_8^{2.3} & W_8^{2.4} & W_8^{2.5} & W_8^{2.6} & W_8^{2.7} \\ W_8^{3.0} & W_8^{3.1} & W_8^{3.2} & W_8^{3.3} & W_8^{3.4} & W_8^{3.5} & W_8^{3.6} & W_8^{3.7} \\ W_8^{4.0} & W_8^{4.1} & W_8^{4.2} & W_8^{4.3} & W_8^{4.4} & W_8^{4.5} & W_8^{4.6} & W_8^{4.7} \\ W_8^{5.0} & W_8^{5.1} & W_8^{5.2} & W_8^{5.3} & W_8^{5.4} & W_8^{5.5} & W_8^{5.6} & W_8^{5.7} \\ W_8^{6.0} & W_8^{6.1} & W_8^{6.2} & W_8^{6.3} & W_8^{6.4} & W_8^{6.5} & W_8^{6.6} & W_8^{6.7} \\ W_8^{7.0} & W_8^{7.1} & W_8^{7.2} & W_8^{7.3} & W_8^{7.4} & W_8^{7.5} & W_8^{7.6} & W_8^{7.7} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ f(6) \\ f(7) \end{bmatrix}$$

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



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

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

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、

$$\begin{aligned} F_{2m} &= \sum_{n=0}^{\frac{N}{2}-1} f(n) W_N^{2mn} + \sum_{n=\frac{N}{2}}^{N-1} f(n) W_N^{2mn} && \text{前半と後半に分割} \\ &= \sum_{n=0}^{\frac{N}{2}-1} f(n) W_N^{2mn} + \sum_{n=0}^{\frac{N}{2}-1} f\left(n + \frac{N}{2}\right) W_N^{2m(n+\frac{N}{2})} && n=n'+N/2と変数変換したうえでn'をnに書き戻す \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) + f\left(n + \frac{N}{2}\right)\right) W_N^{2mn} && \because W_N^{mN} = 1 \text{ から} \\ &= \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) + f\left(n + \frac{N}{2}\right)\right) W_{N/2}^{mn} && \because W_N^{2mn} = \exp\left(-j \frac{2\pi}{N} 2mn\right) = \exp\left(-j \frac{2\pi}{N/2} mn\right) \end{aligned}$$

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

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

k=2m+1、すなわち奇数番目の項は、

$$\begin{aligned} F_{2m+1} &= \sum_{n=0}^{\frac{N}{2}-1} f(n) W_N^{(2m+1)n} + \sum_{n=\frac{N}{2}}^{N-1} f(n) W_N^{(2m+1)n} \\ &= \\ &= \\ &= && \because W_N^{mN} = 1, W_N^{N/2} = -1 \text{ から} \\ &= && \because W_N^{2mn} = \exp\left(-j \frac{2\pi}{N} 2mn\right) = \exp\left(-j \frac{2\pi}{N/2} mn\right) \end{aligned}$$

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



DFTの計算量を削減する

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

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

つまり、**N個**の信号のフーリエ変換（乗算が**N²**個必要）を、
2つの、**N/2個**の信号のフーリエ変換（乗算が**N²/2**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。

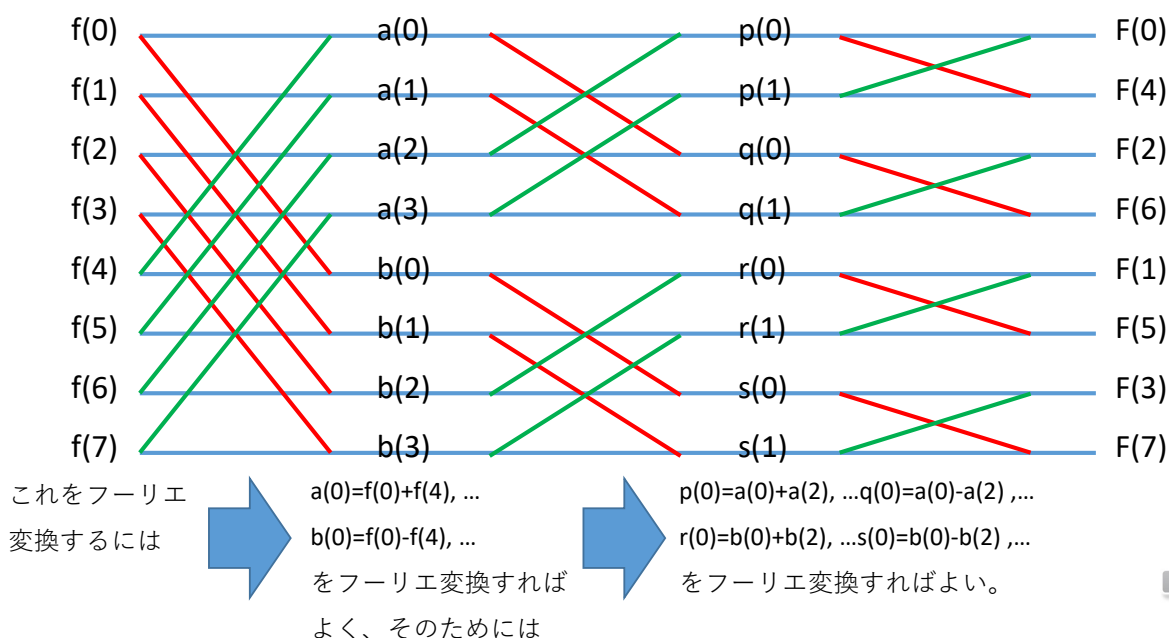
このように、**1/2**分割による計算量の削減を行う手法を
高速フーリエ変換、**FFT(Fast Fourier Transform)**と呼ぶ。



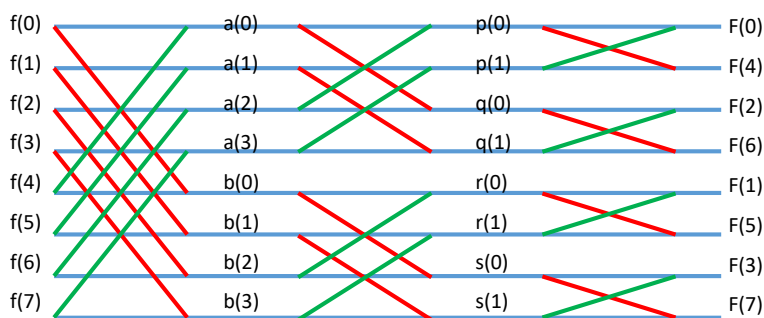
N=8の場合

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

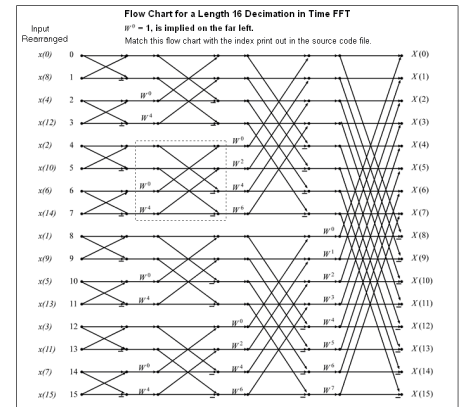
$$F_{2m+1} = \sum_{n=0}^{\frac{N}{2}-1} \left(f(n) - f\left(n + \frac{N}{2}\right) \right) W_N^n W_N^{2mn} \quad \text{N/2個の信号} \left(f(n) - f\left(n + \frac{N}{2}\right) \right) W_N^n \text{のフーリエ変換}$$



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, 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。

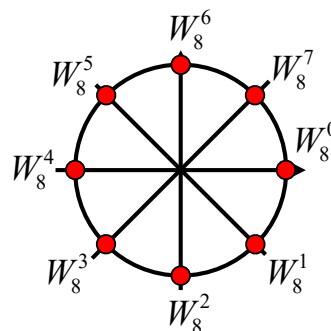
提出締め切り：講義日から一週間以内



DFTの計算量を削減する

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

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



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

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

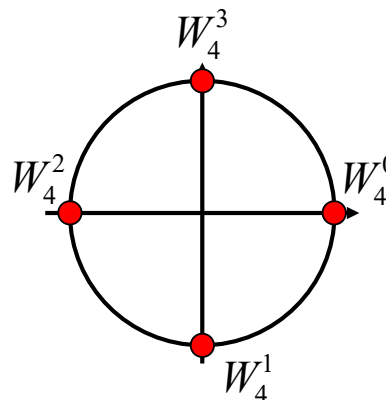
この規則性から、計算量を減らす工夫を見出す。

DFTの計算量を削減する：4点の場合

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

2列目と3列目を入れ替える

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



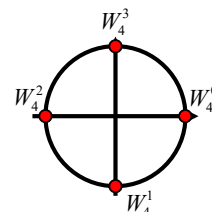
DFTの計算量を削減する

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

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

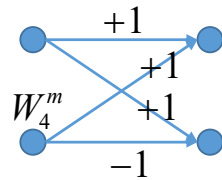
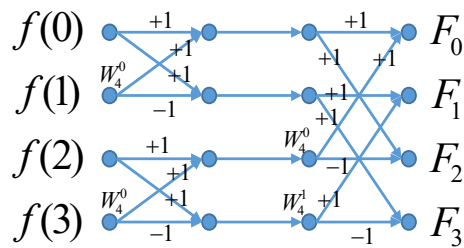


この式には, $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ という配列が 4 つ含まれることに注目

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



バタフライユニット：
乗算回数は1回

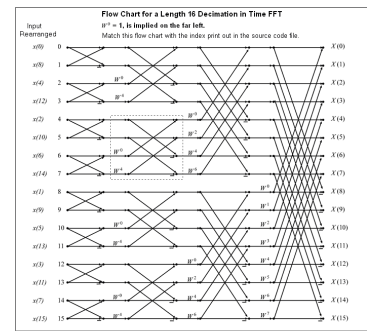
4つのバタフライユニットだから、乗算回数は4回
→もともと $4 \times 4 = 16$ 回だったから1/4に。

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。



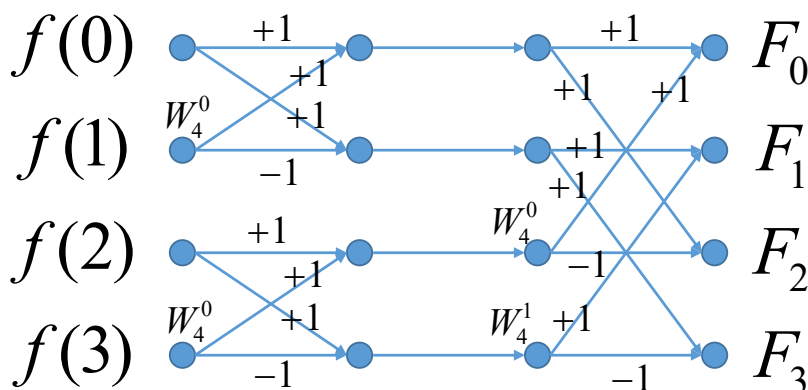
16点の場合

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

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