

デジタル信号処理 (X)

物理情報システム専攻
山口雅浩

E-mail: yamaguchi.m.aa@m.titech.ac.jp

Web: <http://www-oid.ip.titech.ac.jp>

9. 高速フーリエ変換 (FFT)

• N点DFTの計算

$$X(k) = \text{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}$$

– N^2 回の複素乗算

– $N(N-1)$ 回の複素加算

• 高速フーリエ変換(Fast Fourier Transform: FFT)

– DFTの性質を利用して乗算回数を削減

• Cooley and Tukey のアルゴリズム

– $N=2^p$ のときが多く用いられる

→ 乗算回数は $(N/2)(\log_2 N - 1)$

$$X(k) = \text{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}$$

n を偶数と奇数に分ける
 $l = 0, 1, \dots, N/2-1$ を用いて

$$X(k) = \sum_{l=0}^{N/2-1} x(2l) \cdot W_N^{2lk} + \sum_{l=0}^{N/2-1} x(2l+1) \cdot W_N^{(2l+1)k}$$

$$W_N^{2lk} = e^{-j\frac{2\pi}{N}(2lk)} = e^{-j\frac{2\pi}{N/2}lk} = W_{N/2}^{lk} \quad \text{と変形できるので、}$$

$$X(k) = \sum_{l=0}^{N/2-1} x(2l) \cdot W_{N/2}^{lk} + W_N^k \sum_{l=0}^{N/2-1} x(2l+1) \cdot W_{N/2}^{lk}$$

$G(k) \quad H(k)$

$(N/2)$ 点DFTの式と同じだが、 $k = 0, 1, \dots, N-1$

$$G(k) = \sum_{l=0}^{N/2-1} x(2l) \cdot W_{N/2}^{lk}$$

$$H(k) = \sum_{l=0}^{N/2-1} x(2l+1) \cdot W_{N/2}^{lk}$$

を $k = 0, 1, \dots, N-1$ について計算する必要がある

しかし、 $N/2 \leq k \leq N-1$ では、 $k' = k - N/2$ として

$$W_{N/2}^{lk} = W_{N/2}^{l(k'+N/2)} = W_{N/2}^{lk'+(N/2)l} = W_{N/2}^{lk'}$$

が成り立つことから、 $k = N/2, N/2+1, \dots, N-1$ に対しては

$$G(k) = G(k - N/2)$$

$$H(k) = H(k - N/2)$$

であり、 $G(k), H(k)$ の計算は、 $k = 0, 1, \dots, N/2-1$ だけ行えばよい

$(N/2)$ 点DFT $\times 2$

さらに、同様に、 $N/2 \leq k \leq N-1$ では $k' = k - N/2$ として

$$W_N^k = W_N^{k'+N/2} = W_N^{k'} W_N^{N/2} = W_N^{k'} \cdot e^{-j\frac{2\pi}{N} \frac{N}{2}} = W_N^{k'} \cdot e^{-j\pi} = -W_N^{k'}$$

であることから、 W_N^k の計算も $k = 0, 1, \dots, N/2-1$ まででよく、

以下のようにして計算できる

$$k = 0, 1, \dots, N/2-1$$

$$X(k) = G(k) + W_N^k H(k)$$

$$k = N/2, N/2+1, \dots, N-1$$

$$(k' = 0, 1, \dots, N/2-1)$$

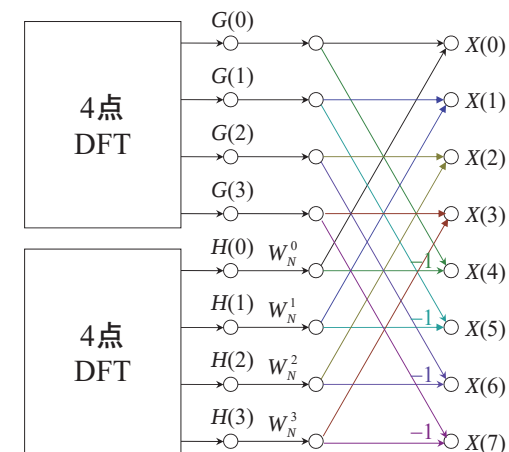
→ 2個の $N/2$ 点DFTで計算できる

$$X(k) = G(k') - W_N^{k'} H(k')$$

$$\left(\frac{N}{2}\right)^2 \times 2 + \frac{N}{2}$$

5

シグナルフローグラフ



7

$G(k), H(k)$ の計算にも同じ方法を適用し、これを繰り返す

$N = 2^p$ とすると、 p 回繰り返し

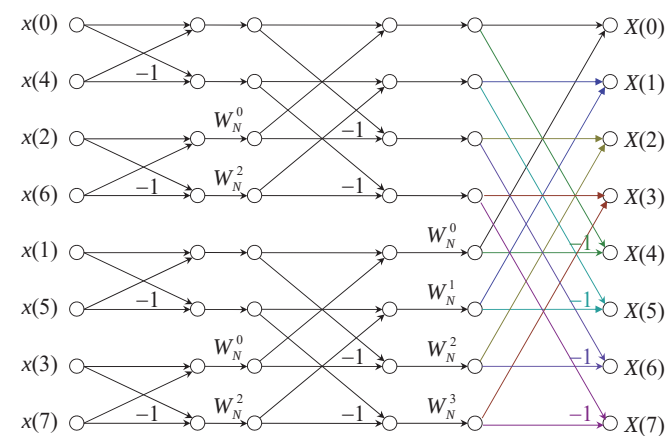
$N = 8 = 2^3$ の例 -- 偶数ならG、奇数ならH

k	1回目	2回目	3回目					k 2進	bit逆順	→10進
		G H	GG HG GH HH							
0	G	G	G				GGG	000	000	0
1	H		G		G		HGG	001	100	4
2	G	H				G	GHG	010	010	2
3	H		H				HHG	011	110	6
4	G	G		H			GGH	100	001	1
5	H		G		H		HGH	101	101	5
6	G	H				H	GHH	110	011	3
7	H		H				HHH	111	111	7

← ここからはじめ

この順に並べ替え

シグナルフローグラフ



→ バタフライ演算

8

計算量について(乗算回数)

2点DFTの計算

$$x(l) + x(l + \frac{N}{2})$$

$$x(l) - x(l + \frac{N}{2})$$

...乗算回数は0

各ステップにおいて

$$W_N^0 \dots W_N^{N/2}$$

の複素乗算 $\rightarrow (N/2)$

(ただし $N=2$ のときは不要)

$\log_2 N$ ステップ



$$\frac{N}{2}(\log_2 N - 1) \text{ or } O\{N \log_2 N\}$$

9

実際に計算してみよう

12/21

離散時間信号 $x(n) = 0, 2, 4, 6, 4, 2, 0, 0$

$n = 0, 1, 2, 3, 4, 5, 6, 7$

において、8点のDFT $X(k)$ を求めよ

まず、並べ替え(右図)

0, 2, 4, 6, 4, 2, 0, 0

$\rightarrow$

0, 4, 4, 0, 2, 2, 6, 0

n	$x(n)$	n のbit 逆順		並べ替えた $x(n)$
0	0	0		0
1	2	4		4
2	4	2		4
3	6	6		0
4	4	1		2
5	2	5		2
6	0	3		6
7	0	7		0

bit逆順 $x(n)$	1段目	係数	2段目	係数	3段目
0	$0+4=4$	1	$4+4=8$	1	$8+10$
4	$0-4=-4$	1	$-4-4j$	1	$-4-4j + \frac{1}{\sqrt{2}}(1-j)(-6j)$
4	$4+0=4$	$W_8^0=1$	$4-4=0$	1	$0-2(-j)$
0	$4-0=4$	$W_8^2=-j$	$-4+4j$	1	$-4+4j - \frac{1}{\sqrt{2}}(1+j) \cdot 6j$
2	$2+2=4$	1	$4+6=10$	$W_8^0=1$	$8-10$
2	$2-2=0$	1	$0-6j=-6j$	$W_8^1 = \frac{1}{\sqrt{2}}(1-j)$	$-4-4j - \frac{1}{\sqrt{2}}(1-j)(-6j)$
6	$6+0=6$	$W_8^0=1$	$4-6=-2$	$W_8^2=-j$	$0+2(-j)$
0	$6-0=6$	$W_8^2=-j$	$0+6j=6j$	$W_8^1 = -\frac{1}{\sqrt{2}}(1+j)$	$-4+4j + \frac{1}{\sqrt{2}}(1+j) \cdot 6j$

これより

$$X(0) = 18$$

$$X(4) = -2$$

$$X(1) = -(3\sqrt{2} + 4)(1+j) \approx -8.2(1+j)$$

$$X(5) = (3\sqrt{2} - 4)(1+j) \approx 0.2(1+j)$$

$$X(2) = 2j$$

$$X(6) = -2j$$

$$X(3) = (3\sqrt{2} - 4)(1-j) \approx 0.2(1-j)$$

$$X(7) = -(3\sqrt{2} + 4)(1-j) \approx -8.2(1-j)$$

FFTを使わないと

$$X(0) = \sum_{k=0}^7 x(k) \cdot e^{-j2\pi 0k/8} = 0 + 2 + 4 + 6 + 4 + 2 + 0 + 0 = 18$$

$$X(1) = 2e^{-j2\pi 1/8} + 4e^{-j2\pi 2/8} + 6e^{-j2\pi 3/8} + 4e^{-j2\pi 4/8} + 2e^{-j2\pi 5/8} =$$

$$X(2) = 2e^{-j2\pi 2/8} + 4e^{-j2\pi 4/8} + 6e^{-j2\pi 6/8} + 4e^{-j2\pi 8/8} + 2e^{-j2\pi 10/8}$$

$$= 2e^{-j2\pi 2/8} + 4e^{-j2\pi 4/8} + 6e^{-j2\pi 6/8} + 4e^{-j2\pi 0/8} + 2e^{-j2\pi 2/8} =$$

$$X(3) = 2e^{-j2\pi 3/8} + 4e^{-j2\pi 6/8} + 6e^{-j2\pi 9/8} + 4e^{-j2\pi 12/8} + 2e^{-j2\pi 15/8}$$

$$= 2e^{-j2\pi 3/8} + 4e^{-j2\pi 6/8} + 6e^{-j2\pi 1/8} + 4e^{-j2\pi 4/8} + 2e^{-j2\pi 7/8} =$$

$\vdots$

$$X(7) = 2e^{-j2\pi 7/8} + 4e^{-j2\pi 14/8} + 6e^{-j2\pi 21/8} + 4e^{-j2\pi 28/8} + 2e^{-j2\pi 35/8} =$$

留意点

- FFT, DFTのプログラムでは、あらかじめ $\cos$, $\sin$ は計算してテーブルとして持っておく
($\cos$, $\sin$ の計算には時間が掛かるため)

- IFFTは



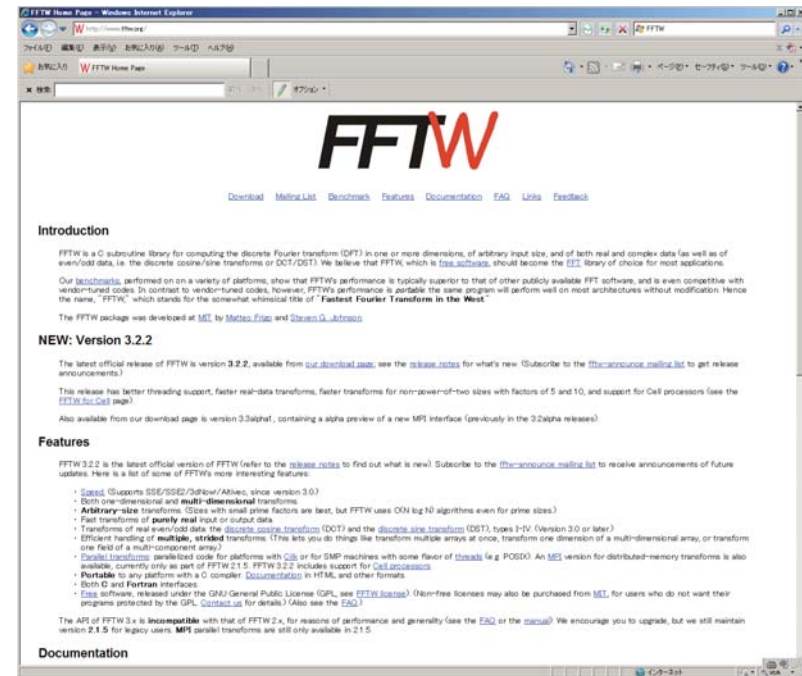
により計算できる。

- 実数であることがわかっている信号のFFTでは、 $X(0) \dots X(N/2)$ まで計算し、 $X(N/2+1) \dots X(N-1)$ を $X(N-k+1)$ の複素共役として求めてもよい。

※ただし一般に $X(k)$ は複素数なのでIFFTには使えない

- FFTのプログラムは書籍・webなどから比較的容易に入手可能。
(FFTは信号処理以外でも幅広く用いられている)

13



www.fftw.org

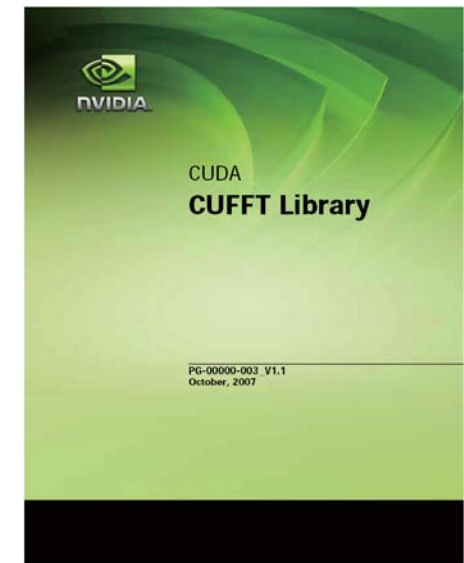
15

さらに深く習いたい人は

- ここで紹介したFFTアルゴリズムは「時間間引き形」と呼ばれるもの。他に「周波数間引き形」のアルゴリズムもある。
ref. 「デジタル信号処理の基礎」、辻井重男監修、電子情報通信学会編など
- D が 2^n では無い場合のアルゴリズムもある。
- 時間信号の周波数解析を行う場合、DFTは有限長の時間信号を周期関数として扱っていることに留意する必要がある。端部の影響を避けるために、「窓関数」がよく用いられる。有名なものにハミング窓、ハニング窓などがある。
- 画像などの2次元信号に合わせたFFTもある。計算時間がそれほど問題にならない場合には、1次元FFTを各行に適用し、その結果の各列に対して再度1次元FFTを適用することで2次元信号のDFTを行うことができる。
- DFTは離散系列に対する直交変換の一種である。DFT以外の直交変換も多数ある。離散コサイン変換(DCT)はJPEGなどの画像圧縮に用いられる。他に、アダマール変換、ハール変換、KL変換などがある。

14

GPUによるFFT



参考:

http://www.nvidia.co.jp/object/cuda_develop_jp.html

<http://www.gsic.titech.ac.jp/~ccwww/tebiki/tesla/tesla6.html>

16