

信号処理とフーリエ変換 第9回

～離散 Fourier 変換 (2)～

かつらだ まさし
桂田 祐史

<http://nalab.mind.meiji.ac.jp/~mk/fourier2021/>

2021年11月24日

目次

1 本日の内容・連絡事項

2 離散 Fourier 変換 (続き)

● 離散 Fourier 変換

- 記号についての約束 インデックスは 0 から
- 離散 Fourier 変換の定義
- 離散 Fourier 変換の表現行列と逆変換
- 複素指数関数の選点直交性
- $(\frac{1}{N}\omega^{-jk})^{-1} = (\omega^{jk})$ の証明
- ユニタリ変換への修正

● 高速 Fourier 変換 (FFT)

3 補足: 線形代数メモ

4 余談

本日の内容・連絡事項

- 前回、離散 Fourier 係数を定義し、サンプリング定理を述べた。今回は、 \mathbb{C}^N から \mathbb{C}^N への写像としての**離散 Fourier 変換の定義**を述べ、その逆変換 (いわゆる**反転公式**) を求める。議論はほぼ純粹の線形代数である。さらに離散 Fourier 変換のアルゴリズムである**高速 Fourier 変換 (FFT)** の紹介をする。講義ノート [1] の§3.2, §3.3に相当する。
- 次回は音声信号の周波数を調べる実験を行う予定 (Mathematica を用いる)。またレポート課題 2 も出す予定である。
- 今回用いるのは線形代数であるが、文字式ベースの話なので分かりにくいかもしれない。補足を用意しておいた。

3.2 離散 Fourier 変換

前項 (§3.1) で、周期関数 f をサンプリングして得られた周期 N の周期数列 $\{f_j\}$ ($f_j = f(x_j)$) に対して、離散フーリエ係数と呼ばれる $\{C_n\}$ を定義した:

$$C_n := \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j \quad (n \in \mathbb{Z})$$

ただし

$$\omega := \exp \frac{2\pi i}{N}.$$

3.2 離散 Fourier 変換

前項 (§3.1) で、周期関数 f をサンプリングして得られた周期 N の周期数列 $\{f_j\}$ ($f_j = f(x_j)$) に対して、離散フーリエ係数と呼ばれる $\{C_n\}$ を定義した:

$$C_n := \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j \quad (n \in \mathbb{Z})$$

ただし

$$\omega := \exp \frac{2\pi i}{N}.$$

数列 $\{C_n\}$ も周期 N の周期数列であることが分かった。周期が N であるから、連続した N 項だけを考えれば良い。

ここでは、その変換 $\mathbb{C}^n \ni (f_0, f_1, \dots, f_{N-1}) \mapsto (C_0, C_1, \dots, C_{N-1}) \in \mathbb{C}^N$ の逆変換の話をする。

3.2 離散 Fourier 変換

前項 (§3.1) で、周期関数 f をサンプリングして得られた周期 N の周期数列 $\{f_j\}$ ($f_j = f(x_j)$) に対して、離散フーリエ係数と呼ばれる $\{C_n\}$ を定義した:

$$C_n := \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j \quad (n \in \mathbb{Z})$$

ただし

$$\omega := \exp \frac{2\pi i}{N}.$$

数列 $\{C_n\}$ も周期 N の周期数列であることが分かった。周期が N であるから、連続した N 項だけを考えれば良い。

ここでは、その変換 $\mathbb{C}^n \ni (f_0, f_1, \dots, f_{N-1}) \mapsto (C_0, C_1, \dots, C_{N-1}) \in \mathbb{C}^N$ の逆変換の話をする。

議論は実質的に線形代数であるので、収束などを考える必要がなく、その意味ではあまり難しくなく、きちんとした話が出る。逆行列を求める話に直交性が効いてくるのが、個人的にはとても面白い。

3.2 離散 Fourier 変換

3.2.1 記号についての約束 インデックスは 0 から

定理を述べる前に、記号についての約束をする。

3.2 離散 Fourier 変換

3.2.1 記号についての約束 インデックスは 0 から

定理を述べる前に、記号についての約束をする。

線形代数では、ベクトルや行列の成分は、1 から番号をつける (行や列の番号は 1 から始める) のが普通だが、ここでは 0 から番号をつけることにする。

3.2 離散 Fourier 変換

3.2.1 記号についての約束 インデックスは 0 から

定理を述べる前に、記号についての約束をする。

線形代数では、ベクトルや行列の成分は、1 から番号をつける (行や列の番号は 1 から始める) のが普通だが、ここでは 0 から番号をつけることにする。

またベクトルの一般の成分を表すのに第 i 成分、行列の一般の成分を表すのに (i, j) 成分を指定することが多いが、 i は虚数単位を表す記号として用いたいため、ここでは、

- ベクトルの一般の成分を表すのに第 j 成分
- 行列の一般の成分を表すのに (j, k) 成分

を指定する。

3.2 離散 Fourier 変換

3.2.1 記号についての約束 インデックスは 0 から

定理を述べる前に、記号についての約束をする。

線形代数では、ベクトルや行列の成分は、1 から番号をつける (行や列の番号は 1 から始める) のが普通だが、ここでは 0 から番号をつけることにする。

またベクトルの一般の成分を表すのに第 i 成分、行列の一般の成分を表すのに (i, j) 成分を指定することが多いが、 i は虚数単位を表す記号として用いたいため、ここでは、

- ベクトルの一般の成分を表すのに第 j 成分
- 行列の一般の成分を表すのに (j, k) 成分

を指定する。

$$\mathbf{x} = (x_j) = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix}, \quad A = (a_{jk}) = \begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0,N-1} \\ a_{10} & a_{11} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N-1,0} & a_{N-1,1} & & a_{N-1,N-1} \end{pmatrix}.$$

(「線形代数の補足」というのをスライド 18,19 ページに書いたため、適宜参照して下さい。)

3.2.2 離散 Fourier 変換の定義

定義 9.1 (離散 Fourier 変換)

$N \in \mathbb{N}$ に対して、 $\omega := e^{2\pi i/N}$ とおく。 $\mathbf{f} = (f_0, f_1, \dots, f_{N-1})^\top \in \mathbb{C}^N$ に対して

$$(1) \quad C_n := \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j \quad (n = 0, 1, \dots, N-1)$$

で定まる $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^\top \in \mathbb{C}^N$ を \mathbf{f} の **離散 Fourier 変換** (discrete Fourier transform) と呼ぶ。また、写像 $\mathcal{F}: \mathbb{C}^N \ni \mathbf{f} \mapsto \mathbf{C} \in \mathbb{C}^N$ のことも **離散 Fourier 変換** と呼ぶ。

3.2.2 離散 Fourier 変換の定義

定義 9.1 (離散 Fourier 変換)

$N \in \mathbb{N}$ に対して、 $\omega := e^{2\pi i/N}$ とおく。 $\mathbf{f} = (f_0, f_1, \dots, f_{N-1})^\top \in \mathbb{C}^N$ に対して

$$(1) \quad C_n := \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j \quad (n = 0, 1, \dots, N-1)$$

で定まる $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^\top \in \mathbb{C}^N$ を \mathbf{f} の **離散 Fourier 変換** (discrete Fourier transform) と呼ぶ。また、写像 $\mathcal{F}: \mathbb{C}^N \ni \mathbf{f} \mapsto \mathbf{C} \in \mathbb{C}^N$ のことも **離散 Fourier 変換** と呼ぶ。

\mathcal{F} が線型写像であることはすぐ分かる:

$$\mathcal{F}(\mathbf{f} + \mathbf{g}) = \mathcal{F}\mathbf{f} + \mathcal{F}\mathbf{g}, \quad \mathcal{F}(c\mathbf{f}) = c\mathcal{F}\mathbf{f} \quad (\mathbf{f}, \mathbf{g} \in \mathbb{C}^N, c \in \mathbb{C}).$$

ゆえに \mathbf{C} は \mathbf{f} にある行列 W をかけることで求まる:

$$\mathcal{F}(\mathbf{f}) = W\mathbf{f} \quad (\mathbf{f} \in \mathbb{C}^N).$$

逆変換 \mathcal{F}^{-1} が存在することが(後で)分かる。もちろん $\mathcal{F}^{-1}(\mathbf{f}) = W^{-1}\mathbf{f}$.

3.2.3 離散 Fourier 変換の表現行列と逆変換

定理 9.2 (離散 Fourier 変換の表現行列とその逆行列)

$N \in \mathbb{N}$ に対して $\omega := e^{2\pi i/N}$, $W := \frac{1}{N} (\omega^{-jk})$ (N 次正方行列) とおくと、

$$\forall \mathbf{f} = (f_j), \quad \forall \mathbf{C} = (C_j) \in \mathbb{C}^N$$

に対して

$$(2) \quad C_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j \omega^{-nj} \quad (n = 0, 1, \dots, N-1) \Leftrightarrow \mathbf{C} = W\mathbf{f}.$$

W は正則で、

$$(3) \quad W^{-1} = (\omega^{jk}).$$

ゆえに次の反転公式が成り立つ。

$$(4) \quad C_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j \omega^{-nj} \quad (0 \leq n \leq N-1) \Leftrightarrow f_j = \sum_{n=0}^{N-1} C_n \omega^{jn} \quad (0 \leq j \leq N-1).$$

3.2.3 離散 Fourier 変換の表現行列と逆変換 行列の表示

念のため、 W や (ω^{jk}) を普通のやり方で書いておく。

$$W = \frac{1}{N} (\omega^{-jk}) = \frac{1}{N} \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(N-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \dots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-(N-1)2} & \dots & \omega^{-(N-1)(N-1)} \end{pmatrix}.$$

また

$$(\omega^{jk}) = \begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{(N-1)2} & \dots & \omega^{(N-1)(N-1)} \end{pmatrix}.$$

1 から番号づけるルールでは、 $W = (\frac{1}{N}\omega^{-(i-1)(j-1)})$, $W^{-1} = (\omega^{(i-1)(j-1)})$.

3.2.3 離散 Fourier 変換の表現行列と逆変換

上の定理を背景に次のように定義する。

定義 9.3 (逆離散 Fourier 変換)

$N \in \mathbb{N}$ に対して $\omega := \exp \frac{2\pi i}{N}$ とおく。 $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^T \in \mathbb{C}^N$ に対して

$$(5) \quad f_j = \sum_{n=0}^{N-1} C_n \omega^{jn} \quad (j = 0, 1, \dots, N-1)$$

で定義される $\mathbf{f} = (f_j)$ を \mathbf{C} の **逆離散 Fourier 変換** (inverse discrete Fourier transform) と呼ぶ。また写像 $\mathbb{C}^N \ni \mathbf{C} \mapsto \mathbf{f} \in \mathbb{C}^N$ も **逆離散 Fourier 変換** と呼ぶ。

3.2.3 離散 Fourier 変換の表現行列と逆変換

上の定理を背景に次のように定義する。

定義 9.3 (逆離散 Fourier 変換)

$N \in \mathbb{N}$ に対して $\omega := \exp \frac{2\pi i}{N}$ とおく。 $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^T \in \mathbb{C}^N$ に対して

$$(5) \quad f_j = \sum_{n=0}^{N-1} C_n \omega^{jn} \quad (j = 0, 1, \dots, N-1)$$

で定義される $\mathbf{f} = (f_j)$ を \mathbf{C} の **逆離散 Fourier 変換** (inverse discrete Fourier transform) と呼ぶ。また写像 $\mathbb{C}^N \ni \mathbf{C} \mapsto \mathbf{f} \in \mathbb{C}^N$ も **逆離散 Fourier 変換** と呼ぶ。

定理 9.2 の証明のうち、(2) は簡単である。

3.2.3 離散 Fourier 変換の表現行列と逆変換

上の定理を背景に次のように定義する。

定義 9.3 (逆離散 Fourier 変換)

$N \in \mathbb{N}$ に対して $\omega := \exp \frac{2\pi i}{N}$ とおく。 $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^T \in \mathbb{C}^N$ に対して

$$(5) \quad f_j = \sum_{n=0}^{N-1} C_n \omega^{jn} \quad (j = 0, 1, \dots, N-1)$$

で定義される $\mathbf{f} = (f_j)$ を \mathbf{C} の **逆離散 Fourier 変換** (inverse discrete Fourier transform) と呼ぶ。また写像 $\mathbb{C}^N \ni \mathbf{C} \mapsto \mathbf{f} \in \mathbb{C}^N$ も **逆離散 Fourier 変換** と呼ぶ。

定理 9.2 の証明のうち、(2) は簡単である。

また (4) は (3) を認めれば明らかである。

3.2.3 離散 Fourier 変換の表現行列と逆変換

上の定理を背景に次のように定義する。

定義 9.3 (逆離散 Fourier 変換)

$N \in \mathbb{N}$ に対して $\omega := \exp \frac{2\pi i}{N}$ とおく。 $\mathbf{C} = (C_0, C_1, \dots, C_{N-1})^T \in \mathbb{C}^N$ に対して

$$(5) \quad f_j = \sum_{n=0}^{N-1} C_n \omega^{jn} \quad (j = 0, 1, \dots, N-1)$$

で定義される $\mathbf{f} = (f_j)$ を \mathbf{C} の **逆離散 Fourier 変換** (inverse discrete Fourier transform) と呼ぶ。また写像 $\mathbb{C}^N \ni \mathbf{C} \mapsto \mathbf{f} \in \mathbb{C}^N$ も **逆離散 Fourier 変換** と呼ぶ。

定理 9.2 の証明のうち、(2) は簡単である。

また (4) は (3) を認めれば明らかである。

問題は、行列 W の正則性と (3) を示すことであるが、それには W と (ω^{jk}) の積を計算して、単位行列に等しいことを示せば良い。その方法がもっとも手短な証明であると思われるが、過去の講義で受講者の反応を見てみるとどうもかんばしくないのので、以下意味がつけられるような証明をいくつか示すことにする。

3.2.4 複素指数関数の選点直交性

補題 9.4 (複素指数関数の選点直交性)

$N \in \mathbb{N}$ に対して、 $\omega := e^{2\pi i/N}$,

$$\varphi_n := (\omega^{nj}) = \begin{pmatrix} \omega^{n \cdot 0} \\ \omega^{n \cdot 1} \\ \vdots \\ \omega^{n \cdot (N-1)} \end{pmatrix} \quad (n = 0, 1, \dots, N-1)$$

とおくと

$$(6) \quad (\varphi_j, \varphi_k) = N\delta_{jk} \quad (j, k = 0, 1, \dots, N-1)$$

が成り立つ。特に $\{\varphi_j\}_{j=0}^{N-1}$ は \mathbb{C}^N の直交系である。

3.2.4 複素指数関数の選点直交性

証明 前回の補題を用いる。

$$\begin{aligned}(\varphi_j, \varphi_k) &= \sum_{n=0}^{N-1} \omega^{nj} \overline{\omega^{nk}} = \sum_{n=0}^{N-1} \omega^{nj} \omega^{-nk} = \sum_{n=0}^{N-1} \omega^{n(j-k)} \\ &= \begin{cases} N & (j - k \equiv 0 \pmod{N}) \\ 0 & (j - k \not\equiv 0 \pmod{N}). \end{cases}\end{aligned}$$

$j, k \in \{0, 1, \dots, N-1\}$ であるから、 $-(N-1) \leq j - k \leq N-1$. ゆえに
 $j - k \equiv 0 \pmod{N} \Leftrightarrow j = k$. ゆえに

$$(\varphi_j, \varphi_k) = N\delta_{jk}. \quad \square$$

3.2.4 複素指数関数の選点直交性

「選点直交性」と呼ばれる理由

$T > 0$, $h := \frac{T}{N}$, $x_j := jh$, $\varphi_n(x) := e^{in\frac{2\pi}{T}x}$ とするとき、

$$\varphi_n(x_j) = e^{in\frac{2\pi}{T}x_j} = e^{2\pi inj/N} = \omega^{nj}.$$

3.2.4 複素指数関数の選点直交性

「選点直交性」と呼ばれる理由

$T > 0$, $h := \frac{T}{N}$, $x_j := jh$, $\varphi_n(x) := e^{in\frac{2\pi}{T}x}$ とするとき、

$$\varphi_n(x_j) = e^{in\frac{2\pi}{T}x_j} = e^{2\pi inj/N} = \omega^{nj}.$$

つまり、 φ_n は、関数 φ_n の標本点 x_0, \dots, x_{N-1} での値を並べたベクトルである:

$$(7) \quad \varphi_n := (\varphi_n(x_0), \varphi_n(x_1), \dots, \varphi_n(x_{N-1}))^T \quad (n = 0, 1, \dots, N-1).$$

(φ_n は φ_n の離散化あるいは標本化と呼ぶのがふさわしい。)

3.2.4 複素指数関数の選点直交性

「選点直交性」と呼ばれる理由

$T > 0$, $h := \frac{T}{N}$, $x_j := jh$, $\varphi_n(x) := e^{in\frac{2\pi}{T}x}$ とするとき、

$$\varphi_n(x_j) = e^{in\frac{2\pi}{T}x_j} = e^{2\pi inj/N} = \omega^{nj}.$$

つまり、 φ_n は、関数 φ_n の標本点 x_0, \dots, x_{N-1} での値を並べたベクトルである:

$$(7) \quad \varphi_n := (\varphi_n(x_0), \varphi_n(x_1), \dots, \varphi_n(x_{N-1}))^T \quad (n = 0, 1, \dots, N-1).$$

(φ_n は φ_n の離散化あるいは標本化と呼ぶのがふさわしい。)

関数 φ, ψ の内積を

$$(\varphi, \psi) = \int_{-T/2}^{T/2} \varphi(x) \overline{\psi(x)} dx$$

で定めるとき、 $\{\varphi_n\}_{n=0}^{N-1}$ は

$$(\varphi_j, \varphi_k) = T\delta_{jk} \quad (j, k \in \mathbb{Z})$$

を満たす直交 (関数) 系であることは学んである。その離散化がやはり直交性 (ただしベクトルとして) を持っているわけである。

(6) は離散直交性とでも言いたくなるけれど、普通は選点直交性と呼ぶらしい。(英語だと、discrete orthogonality という。)

3.2.4 複素指数関数の選点直交性 W の列ベクトルの直交性

系 9.5 (W の列ベクトルの直交性)

$N \in \mathbb{N}$, $\omega := e^{2\pi i/N}$, $W := (\frac{1}{N}\omega^{-jk})$ とする。 W の第 k 列を \mathbf{w}_k とするとき

$$(\mathbf{w}_j, \mathbf{w}_k) = \frac{1}{N}\delta_{jk}$$

が成り立つ。

系 9.5 (W の列ベクトルの直交性)

$N \in \mathbb{N}$, $\omega := e^{2\pi i/N}$, $W := (\frac{1}{N}\omega^{-jk})$ とする。 W の第 k 列を \mathbf{w}_k とするとき

$$(\mathbf{w}_j, \mathbf{w}_k) = \frac{1}{N}\delta_{jk}$$

が成り立つ。

証明

$$\mathbf{w}_k = \frac{1}{N} \begin{pmatrix} \omega^{-0 \cdot k} \\ \omega^{-1 \cdot k} \\ \vdots \\ \omega^{-(N-1) \cdot k} \end{pmatrix} = \frac{1}{N} \boldsymbol{\varphi}_k \quad (\text{複素共役ベクトル})$$

であるから

$$(\mathbf{w}_j, \mathbf{w}_k) = \left(\frac{1}{N} \boldsymbol{\varphi}_j, \frac{1}{N} \boldsymbol{\varphi}_k \right) = \frac{1}{N^2} \overline{(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k)} = \frac{1}{N^2} \cdot \overline{N\delta_{jk}} = \frac{1}{N} \delta_{jk}. \quad \square$$

3.2.5 $\left(\frac{1}{N}\omega^{-jk}\right)^{-1} = (\omega^{jk})$ の証明

証明 1 (列ベクトルが直交系である行列の逆行列を求める定跡手順)
 W と W の Hermite 共役 W^* の積を作ってみると対角行列になる。実際

$$(8) \quad W^* W = \begin{pmatrix} \mathbf{w}_0^* \\ \mathbf{w}_1^* \\ \vdots \\ \mathbf{w}_{N-1}^* \end{pmatrix} (\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_{N-1}) = (\mathbf{w}_j^* \mathbf{w}_k) = \left(\frac{1}{N} \delta_{jk} \right) = \frac{1}{N} I.$$

3.2.5 $\left(\frac{1}{N}\omega^{-jk}\right)^{-1} = (\omega^{jk})$ の証明

証明 1 (列ベクトルが直交系である行列の逆行列を求める定跡手順)
 W と W の Hermite 共役 W^* の積を作ってみると対角行列になる。実際

$$(8) \quad W^*W = \begin{pmatrix} \mathbf{w}_0^* \\ \mathbf{w}_1^* \\ \vdots \\ \mathbf{w}_{N-1}^* \end{pmatrix} (\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_{N-1}) = (\mathbf{w}_j^* \mathbf{w}_k) = \left(\frac{1}{N}\delta_{jk}\right) = \frac{1}{N}I.$$

ゆえに $(NW^*)W = I$ であるから、

$$W^{-1} = NW^* = N \left(\overline{\frac{1}{N}\omega^{-kj}} \right) = (\omega^{kj}) = (\omega^{jk}). \quad \square$$

3.2.5 $(\frac{1}{N}\omega^{-jk})^{-1} = (\omega^{jk})$ の証明 2

任意の $\mathbf{C} \in \mathbb{C}^N$ が与えられたとき

$$\mathbf{C} = W\mathbf{f}$$

を満たす $\mathbf{f} \in \mathbb{C}^N$ を求めよう。

3.2.5 $(\frac{1}{N}\omega^{-jk})^{-1} = (\omega^{jk})$ の証明 2

任意の $\mathbf{C} \in \mathbb{C}^N$ が与えられたとき

$$\mathbf{C} = W\mathbf{f}$$

を満たす $\mathbf{f} \in \mathbb{C}^N$ を求めよう。

\mathbf{C} は $\{\mathbf{w}_n\}$ の線型結合である。実際

$$\mathbf{C} = W\mathbf{f} = (\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_{N-1}) \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = \mathbf{w}_1 f_1 + \mathbf{w}_2 f_2 + \cdots + \mathbf{w}_{N-1} f_{N-1} = \sum_{j=0}^{N-1} f_j \mathbf{w}_j.$$

3.2.5 $(\frac{1}{N}\omega^{-jk})^{-1} = (\omega^{jk})$ の証明 2

任意の $\mathbf{C} \in \mathbb{C}^N$ が与えられたとき

$$\mathbf{C} = W\mathbf{f}$$

を満たす $\mathbf{f} \in \mathbb{C}^N$ を求めよう。

\mathbf{C} は $\{\mathbf{w}_n\}$ の線型結合である。実際

$$\mathbf{C} = W\mathbf{f} = (\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_{N-1}) \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = \mathbf{w}_1 f_1 + \mathbf{w}_2 f_2 + \cdots + \mathbf{w}_{N-1} f_{N-1} = \sum_{j=0}^{N-1} f_j \mathbf{w}_j.$$

ゆえに (直交系による展開の係数を求める公式を適用して)

$$f_j = \frac{(\mathbf{C}, \mathbf{w}_j)}{(\mathbf{w}_j, \mathbf{w}_j)} = \frac{\sum_{k=0}^{N-1} C_k \overline{\frac{1}{N}\omega^{-kj}}}{1/N} = \sum_{k=0}^{N-1} \omega^{jk} C_k.$$

3.2.5 $(\frac{1}{N}\omega^{-jk})^{-1} = (\omega^{jk})$ の証明2

任意の $\mathbf{C} \in \mathbb{C}^N$ が与えられたとき

$$\mathbf{C} = W\mathbf{f}$$

を満たす $\mathbf{f} \in \mathbb{C}^N$ を求めよう。

\mathbf{C} は $\{\mathbf{w}_n\}$ の線型結合である。実際

$$\mathbf{C} = W\mathbf{f} = (\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_{N-1}) \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-1} \end{pmatrix} = \mathbf{w}_1 f_1 + \mathbf{w}_2 f_2 + \cdots + \mathbf{w}_{N-1} f_{N-1} = \sum_{j=0}^{N-1} f_j \mathbf{w}_j.$$

ゆえに (直交系による展開の係数を求める公式を適用して)

$$f_j = \frac{(\mathbf{C}, \mathbf{w}_j)}{(\mathbf{w}_j, \mathbf{w}_j)} = \frac{\sum_{k=0}^{N-1} C_k \overline{\frac{1}{N}\omega^{-kj}}}{1/N} = \sum_{k=0}^{N-1} \omega^{jk} C_k.$$

これは $W^{-1} = (\omega^{jk})$ であることを示している。

□

3.2.6 ユニタリ変換への修正

注意 9.6 (定義を少し修正するとユニタリ変換になる)

$$C_n = \frac{1}{N} \sum_{j=0}^{N-1} \omega^{-nj} f_j, \quad f_j = \sum_{n=0}^{N-1} \omega^{nj} C_n$$

を

$$C'_n = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{-nj} f_j, \quad f_j = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \omega^{nj} C'_n$$

のように変えると、対応する行列は

$$U = \left(\frac{1}{\sqrt{N}} \omega^{-jk} \right), \quad U^{-1} = \left(\frac{1}{\sqrt{N}} \omega^{jk} \right)$$

となる。 $U^* = U^{-1}$ が成り立つ。つまり U はユニタリ行列である。
これは U の列ベクトルが正規直交系ということである。 □

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

ここでは具体的なアルゴリズムの説明は省略する (興味があれば大浦 [3], [4] を見よ)。

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

ここでは具体的なアルゴリズムの説明は省略する (興味があれば大浦 [3], [4] を見よ)。

いくつか注意点を述べておく。

- FFT は近似ではなく、離散 Fourier 係数そのものを計算する。使わない理由はない。(離散 Fourier 係数は、Fourier 係数の近似であるが、それとは別の話。)

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

ここでは具体的なアルゴリズムの説明は省略する (興味があれば大浦 [3], [4] を見よ)。

いくつか注意点を述べておく。

- FFT は近似ではなく、離散 Fourier 係数そのものを計算する。使わない理由はない。(離散 Fourier 係数は、Fourier 係数の近似であるが、それとは別の話。)
- 項数 N が “たくさん” の素因数の積に分解できるときに高速化される。典型的なのは $N = 2^m$ ($m \in \mathbb{N}$) の場合で、そのとき計算量は $O(N \log N)$ である。定義式から素朴に計算すると $O(N^2)$ となるので、その差は大きい。

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

ここでは具体的なアルゴリズムの説明は省略する (興味があれば大浦 [3], [4] を見よ)。

いくつか注意点を述べておく。

- FFT は近似ではなく、離散 Fourier 係数そのものを計算する。使わない理由はない。(離散 Fourier 係数は、Fourier 係数の近似であるが、それとは別の話。)
- 項数 N が “たくさん” の素因数の積に分解できるときに高速化される。典型的なのは $N = 2^m$ ($m \in \mathbb{N}$) の場合で、そのとき計算量は $O(N \log N)$ である。定義式から素朴に計算すると $O(N^2)$ となるので、その差は大きい。
- 現在では、ソフトウェアで離散 Fourier 変換をするとき、FFT が使われると期待できるが、 N の値の選択には注意すべきである (素数など選んではいけない)。

3.3 高速 Fourier 変換 (FFT)

離散 Fourier 変換には、非常に効率の高いアルゴリズムが存在する。それを**高速 Fourier 変換** (the fast Fourier transform) と呼び、**FFT** と略記する。

FFT が広く知られるようになったきっかけは、1965 年の Cooley-Tukey [2] とされるが、それ以前から色々な人達が気づいて使っていたとのことである。

ここでは具体的なアルゴリズムの説明は省略する (興味があれば大浦 [3], [4] を見よ)。

いくつか注意点を述べておく。

- FFT は近似ではなく、離散 Fourier 係数そのものを計算する。使わない理由はない。(離散 Fourier 係数は、Fourier 係数の近似であるが、それとは別の話。)
- 項数 N が “たくさん” の素因数の積に分解できるときに高速化される。典型的なのは $N = 2^m$ ($m \in \mathbb{N}$) の場合で、そのとき計算量は $O(N \log N)$ である。定義式から素朴に計算すると $O(N^2)$ となるので、その差は大きい。
- 現在では、ソフトウェアで離散 Fourier 変換をするとき、FFT が使われると期待できるが、 N の値の選択には注意すべきである (素数など選んではいけない)。
- (重要) 離散 Fourier 変換は、離散 Fourier 係数を求めてくれるが、離散 Fourier 係数から等分点上の関数値を求めるのは、逆離散 Fourier 変換であり、それも同じアルゴリズムで高速化される。

線形代数メモ (1)

$\mathbb{K} = \mathbb{R}$ または \mathbb{C} とする。

成分が \mathbb{K} に属する $m \times n$ 行列全体を $\mathbb{K}^{m \times n}$ と書くことにする ($M(m, n; \mathbb{K})$ のような記号を使う人も多い)。 n 次元縦ベクトルは $n \times 1$ 行列とみなす (つまり $\mathbb{K}^{n \times 1} = \mathbb{K}^n$)。

これはこの §3 の特別ルール

$\mathbf{x} \in \mathbb{K}^n$ の成分は (特に断らない限り) x_j ($j = 0, 1, \dots, n-1$) とする。

$A \in \mathbb{K}^{m \times n}$ の成分は (特に…) a_{jk} ($j = 0, 1, \dots, m-1; k = 0, 1, \dots, n-1$) とする。

行列 $A = (a_{jk}) \in \mathbb{K}^{m \times n}$ に対して、 $(a_{kj}) \in \mathbb{K}^{n \times m}$ を A の転置行列と呼び A^T で表す。また $(\overline{a_{kj}}) \in \mathbb{K}^{n \times m}$ を A の Hermite 共役行列と呼び A^* で表す。

例えば

$$A = \begin{pmatrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \end{pmatrix}$$

とブロック分けされているとき、

$$A^* = \begin{pmatrix} A_{00}^* & A_{10}^* \\ A_{01}^* & A_{11}^* \\ A_{02}^* & A_{12}^* \end{pmatrix}.$$

(これを $W = (\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{N-1})$ について用いた。スライド 26 ページ。)

線形代数メモ (2)

$A = (a_{jk}) \in \mathbb{K}^{\ell \times m}$ と $B = (b_{jk}) \in \mathbb{K}^{m \times n}$ の積 $AB \in \mathbb{K}^{\ell \times n}$ を

$$AB = \left(\sum_{p=0}^{m-1} a_{jp} b_{pk} \right) \quad (AB \text{ の第 } (j, k) \text{ 成分は } \sum_{p=0}^{m-1} a_{jp} b_{pk})$$

で定義する。特に $A \in \mathbb{K}^{N \times N}$ と $\mathbf{x} \in \mathbb{K}^N$ の積は

$$(9) \quad A\mathbf{x} = \left(\sum_{k=0}^{N-1} a_{jk} x_k \right) \quad (A\mathbf{x} \text{ の第 } j \text{ 成分は } \sum_{k=0}^{N-1} a_{jk} x_k)$$

((9) を $\mathbf{C} = W\mathbf{f}$ についての議論で用いた。 $C_n = \frac{1}{N} \sum_{j=0}^{N-1} f_j \omega^{-nj}$ をじっと見てみること。)

写像 $f: \mathbb{K}^n \rightarrow \mathbb{K}^m$ が線型 \Leftrightarrow ある $A \in \mathbb{K}^{m \times n}$ が存在して $f(\mathbf{x}) = A\mathbf{x}$ ($\mathbf{x} \in \mathbb{K}^n$)。

\mathbb{K}^n における内積は次式で定義される:

$$(\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{n-1} x_j \bar{y}_j \quad (\mathbb{K} = \mathbb{R} \text{ のときは単に } (\mathbf{x}, \mathbf{y}) = \sum_{j=0}^{n-1} x_j y_j).$$

次の式が成り立つ (案外よく使う式。今回は $\mathbf{w}_j^* \mathbf{w}_k = (\mathbf{w}_k, \mathbf{w}_j) = \frac{1}{N} \delta_{kj}$ で使った。)

$$(\mathbf{x}, \mathbf{y}) = \mathbf{y}^* \mathbf{x} \quad (\text{右辺は } 1 \times n \text{ 行列 } \mathbf{y}^* \text{ と } n \times 1 \text{ 行列 } \mathbf{x} \text{ の積とみなす}).$$

今回は使っていないが $(A\mathbf{x}, \mathbf{y}) = (\mathbf{x}, A^* \mathbf{y})$ の証明などが分かりやすくなる。

今回の講義の内容は、個々の式のレベルでは良く知られたことがほとんどで、多くの本に載っている。

どういう順に並べれば話が分かりやすくなるか、毎年頭をひねっている。意外に感じるかも知れないが、結構時間を使っている。

逆変換 (反転公式) については、現時点では次のように考えている。

結論を知れば、単純計算で

$$\frac{1}{N} (\omega^{-jk}) (\omega^{jk}) = 1$$

が確認できて、それが逆行列であることの証明となるが、それはやはり下手な説明なのだろう。やはり意味が分かって頭に残る形の説明が望ましい。

色々な説明が可能である。2つの証明を紹介したが、以下の説明も良さそうと思っている。

Fourier 級数に現れる指数関数系 $\varphi_n(x) = e^{in\frac{2\pi}{T}x}$ ($n \in \mathbb{Z}$) は内積空間 $L^2(0, T)$ の直交系で、その標本化 $\varphi_n := (\varphi_n(x_0), \varphi_n(x_1), \dots, \varphi_n(x_{N-1}))^\top = (\omega^{jn})$ ($0 \leq n \leq N-1$) は \mathbb{C}^N の直交系である ($(\varphi_m, \varphi_n) = N\delta_{mn}$ … 選点直交性)。

(次のスライドに続く)

$A := (\varphi_0, \varphi_1, \dots, \varphi_{N-1})$ とおく。 $A = (\omega^{jk})$ である。 $A^{-1} = W$ を示せば、 $W^{-1} = A$ が分かる。

$$\mathbf{b} = A\mathbf{C}$$

は

$$\mathbf{b} = \sum_{j=0}^{N-1} C_j \varphi_j$$

という式に書き直せる。 \mathbf{C} を直交系 $\{\varphi_n\}$ で展開していて、 C_j はこの展開の係数である、ということなので、 C_j は例の公式で求まる。

$$C_j = \frac{(\mathbf{C}, \varphi_j)}{(\varphi_j, \varphi_j)} = \frac{\varphi_j^* \mathbf{C}}{N} = \frac{1}{N} \sum_{n=0}^{N-1} \omega^{-jn} C_n.$$

ゆえに

$$A^{-1} = \left(\frac{1}{N} \omega^{-jk} \right) = W.$$

参考文献

- [1] 桂田祐史：「信号処理とフーリエ変換」講義ノート, <http://nalab.mind.meiji.ac.jp/~mk/fourier/fourier-lecture-notes.pdf>, 以前は「画像処理とフーリエ変換」というタイトルだったのを直した。(2014～).
- [2] Cooley, J. W. and Tukey, J. W.: An Algorithm for the Machine Calculation of Complex Fourier Series, *Mathematics of Computation*, Vol. 19, No. 90, pp. 297–301 (1965), <http://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf> で公開されている。
- [3] 大浦拓哉：高速 Fourier 変換の概略メモ, http://www.kurims.kyoto-u.ac.jp/~ooura/fftman/fft_note_s.pdf (2005).
- [4] 大浦拓哉：FFT (高速フーリエ・コサイン・サイン変換) の概略と設計法, <http://www.kurims.kyoto-u.ac.jp/~ooura/fftman/> …テキスト・エンコーディングが日本語 EUC になっている (1997～).