情報工学科 講義 ディジタル信号処理A (2025/04/17)

第2回
フーリエ級数展開・フーリエ変換から離散フーリエ変換へ

情報工学科 准教授　高道 慎之介

Takamichi Laboratory
慶應義塾大学 高道研究室

ディジタル信号処理Aの授業予定 (仮)

応用数学の復習

第XX回	日付	内容（順次変わっていくので予想）
第01回	2025/04/10	イントロダクション，ディジタル信号処理
第02回	**2025/04/17**	**フーリエ級数展開・フーリエ変換から離散フーリエ変換へ**
第03回	2025/04/24	ラプラス変換からｚ変換へ
第04回	2025/05/01	インパルス応答と伝達関数，安定性
第05回	2025/05/08	ディジタルフィルタ
第06回	2025/05/15	高速フーリエ変換と短時間フーリエ変換
第07回	2025/05/22	総合演習．期末試験の練習としての立ち位置．
期末試験	2025/06/??	(日程は後日アナウンス)

昨年のBの途中まで

前回の課題の解答

以下の課題についてレポートを作成し，LMS 上で提出せよ。

1. サンプリング周波数を 250 Hz とする．750 サンプル (点) のディジタル信号の時間長は何秒か．

2. ディジタル信号からアナログ信号を復元するプログラムがある．
   この抜けている箇所を穴埋めし，プログラムと，復元できる旨を示す図を示せ．

3. x = [3, 0, 1, 0, 2], h = [4, 2, 1] とする．これらの畳み込み結果を示せ．

サンプリング周波数を 250 Hz とする．
750 サンプル (点) のディジタル信号の時間長は何秒か．

- 250 サンプル / 秒である．故に
    - 750 [サンプル] / 250 [サンプル/秒] = 3 [秒]

ディジタル信号からアナログ信号を復元するプログラムがある．この抜けている箇所を穴埋めし，プログラムと，復元できる旨を示す図を示せ

```
import math
import matplotlib.pyplot as plt

T = 0.1 # sampling period [sec]
n_points = 20 # number of points of the digital signal

# analog signal: sin(2 * pi * t)
x_t = lambda t: math.sin(2 * math.pi * t)

# digital signal: x[n] = x(n * T), n = 0, 1, …, n_points-1
x_n = [x_t(n * T) for n in range(n_points)]

def reconst_x(t: float) -> float:
   def _sinc(z: float) -> float: # sinc function
       return 1 if abs(z) < 1e-10 else math.sin(z) / z

   x = 0
   for n in range(n_points):
      x += x_n[n] * _sinc(math.pi * (t / T - n))

   return x

# plot the signals
t_plot = [n * T / 10 for n in range(n_points * 10)]
plt.plot(t_plot, [x_t(t) for t in t_plot], '-', label='Original signal')
plt.stem([n * T for n in range(n_points)], x_n, "r", label='Digital signal')
plt.plot(t_plot, [reconst_x(t) for t in t_plot], '--', color="k", label='Reconstructed signal')
plt.xlim(0, n_points * T)
plt.ylim(-1.1, 1.1)
plt.xlabel("t [sec]")
plt.grid()
plt.legend()
#plt.savefig("reconstruction.png") # uncomment to save the figure
```

x = [3, 0, 1, 0, 2], h = [4, 2, 1] とする．これらの畳み込み結果を示せ．

h[n]	4	2	1				
x[n]	3	0	1	0	2		
x[n-0]	3	0	1	0	2		
x[n-1]		3	0	1	0	2	
x[n-2]			3	0	1	0	2
h[0]x[n-0]	12	0	4	0	8		
h[1]x[n-1]		6	0	2	0	4	
h[2]x[n-2]			3	0	1	0	2
y[n]	12	6	7	2	9	4	2

本日の内容

Do not edit
How to change the design

Join at slido.com
#3642640

Presenting with animations, GIFs or speaker notes? Enable our Chrome extension

slido

フーリエ級数展開とフーリエ変換の違いって何？

フーリエ変換と離散フーリエ変換の違いって何？

本日の内容

- フーリエ級数展開 (Fourier series expansion)
- フーリエ変換 (Fourier transformation)
- 離散フーリエ変換 (Discrete Fourier transformation)

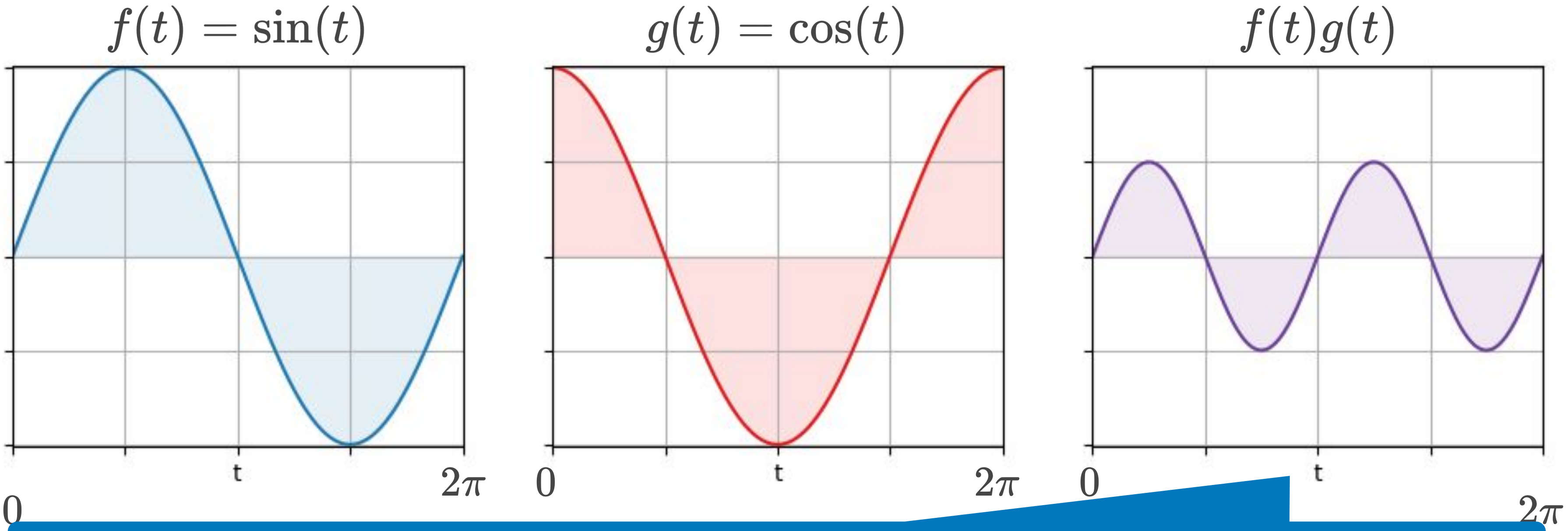
フーリエ級数展開とフーリエ変換を通して，離散フーリエ変換を理解しよう
(応用数学で学習した前提で，少し飛ばしながら復習)

フーリエ級数展開
(Fourier series expansion)

“区間 [a, b] で関数 f(t) と g(t) が直交する” とは

$$\int_a^b f(t)g(t)dt = 0$$

(実数関数の内積)

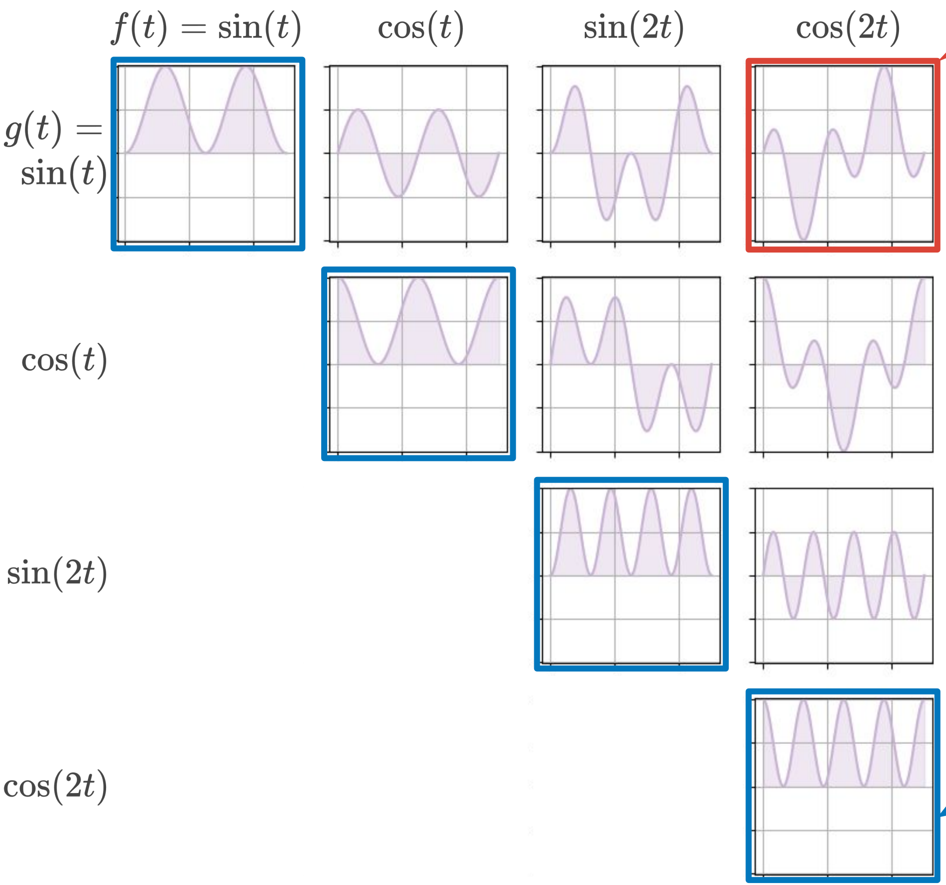


積分すると 0 (塗りつぶしが打ち消しあう) ので，区間 [0, 2π] で sin(t) と cos(t) は直交

(1周期分であればよいので，例えば区間 [-π, π] でも直交)

正弦波関数 {sin(kt), cos(kt)} (k=0, 1, ...) は直交関数系の一種

“直交関数系 = 直交する関数の集まり”と覚えても支障は無い



自分以外と積分すると 直交

$$\int_0^{2\pi} \sin(kt)\sin(lt)dt \quad (k \neq l)$$

$$= -\frac{1}{2}\int_0^{2\pi} \{\underbrace{\cos((k+l)x)}_{\text{積分して 0}} - \underbrace{\cos((k-l)x)}_{\text{積分して 0}}\}\, dx$$

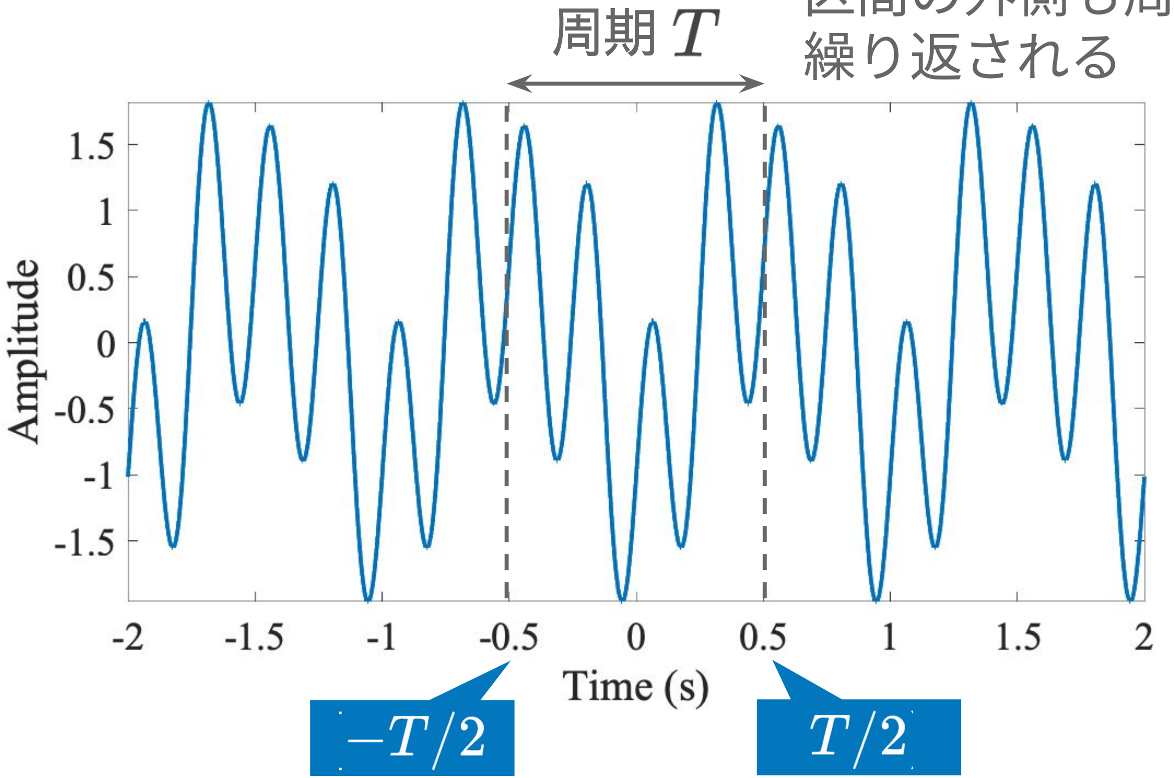
$$= 0$$

- sin を cos にしても同様に証明可能
- k = l の場合は，積分の中身が非負
  - → 積分値は自明に非零

自分自身と積分すると 非零
(この積分値が 1 のとき**正規**直交関数系)

フーリエ級数展開は**周期 T の連続時間信号**を対象にする

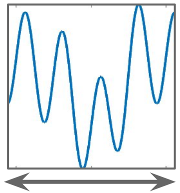
区間の外側も周期 T で
繰り返される



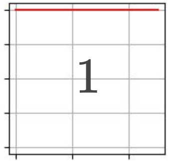
区間の取り方は任意性があり，0 から T でもよい．

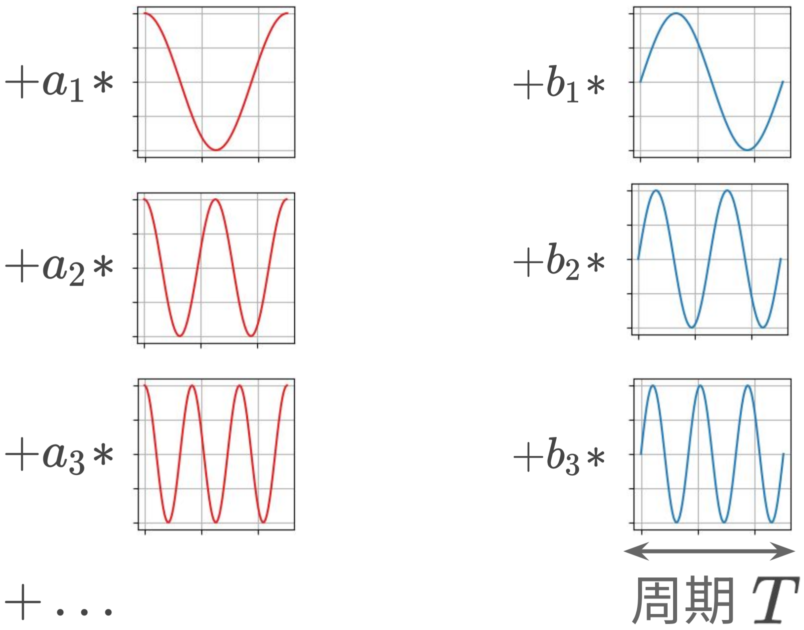
フーリエ級数展開 (**正弦波関数**を基底関数に使う場合)

$$x(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos\left(\frac{2\pi kt}{T}\right) + \sum_{k=1}^{\infty} b_k \sin\left(\frac{2\pi kt}{T}\right)$$



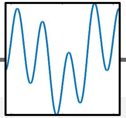
$= a_0 *$

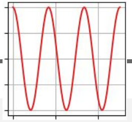




直交関数系を用いて展開．
その係数を “級数” と呼ぶ．

係数の求め方

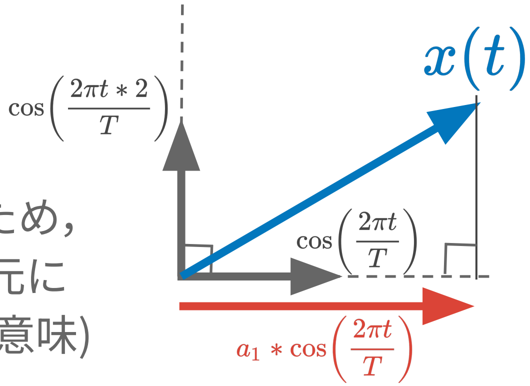




$$a_0 = \frac{1}{T}\int_{-T/2}^{T/2} x(t)dt \quad a_k = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos\left(\frac{2\pi kt}{T}\right)dt \quad b_k = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\sin\left(\frac{2\pi kt}{T}\right)dt$$

- この式の意味するところは…
  - 関数と基底関数の積分 (内積) である．
    - 基底が複素数関数の場合は基底関数の共役
    - 直交関数系の場合はこれが成り立つ
  - 積分の前に定数項がある
    - 基底関数自身と積分 (内積) すると 1 にならないため，その大きさを打ち消している．(変換と逆変換で元に戻らず定数倍されるため定数分の1する，と同じ意味)
    - 正規直交関数系の場合は現れない (内積すると 1)

ベクトルのように考えると分かりやすい

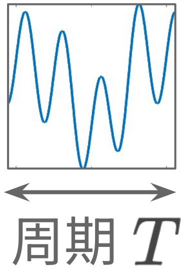


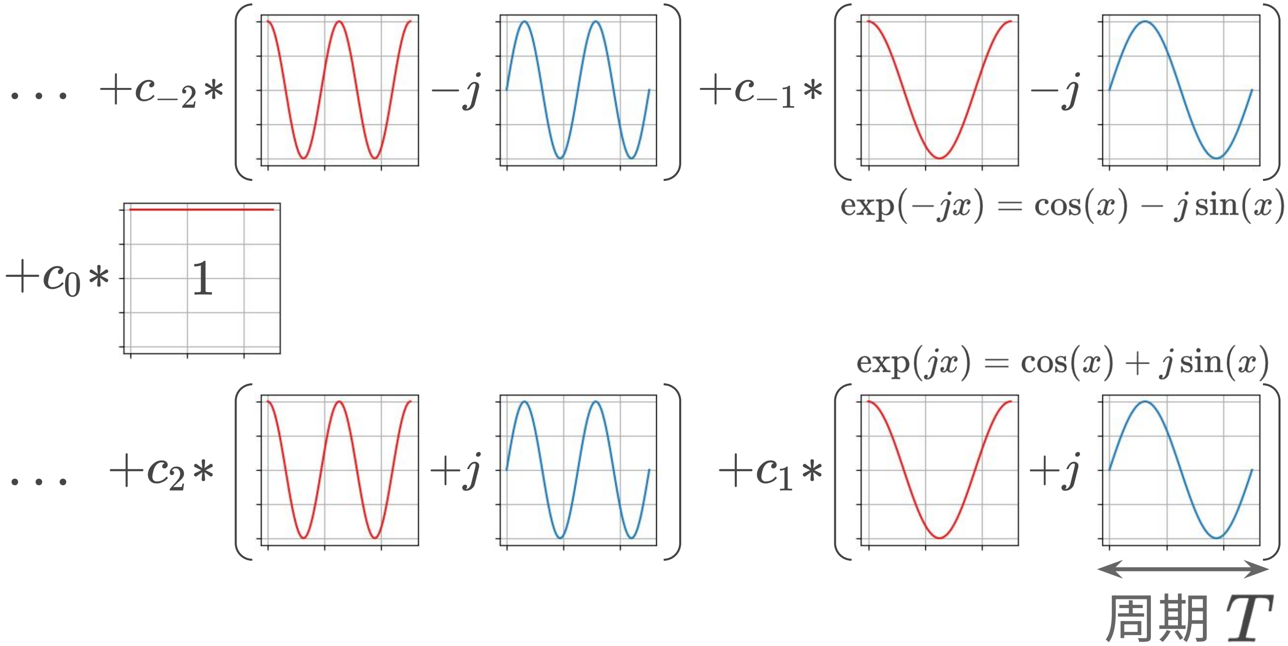
基底との内積が係数

フーリエ級数展開 (**複素指数関数**を基底関数に使う場合)

複素数

$$x(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(j\frac{2\pi kt}{T}\right) \qquad c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(-j\frac{2\pi kt}{T}\right)dt$$





k < 0

k = 0

k > 0

正弦波の係数と対応

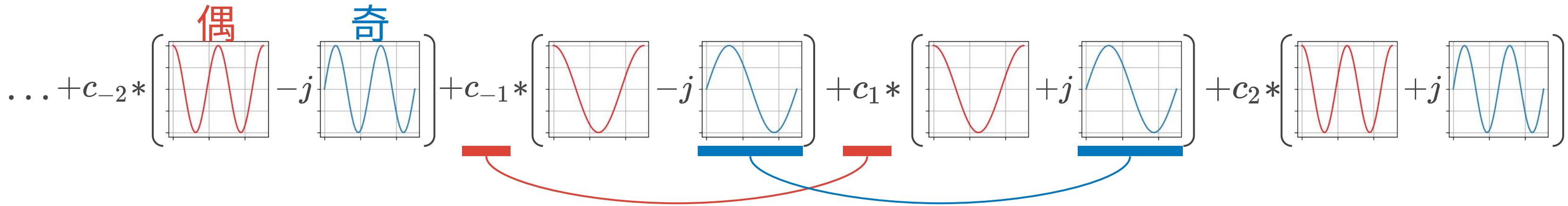
$$\begin{pmatrix} c_0 = a_0 \\ c_k = \dfrac{a_k - jb_k}{2} \end{pmatrix}$$

証明は省略

(応用数学の資料を参照)

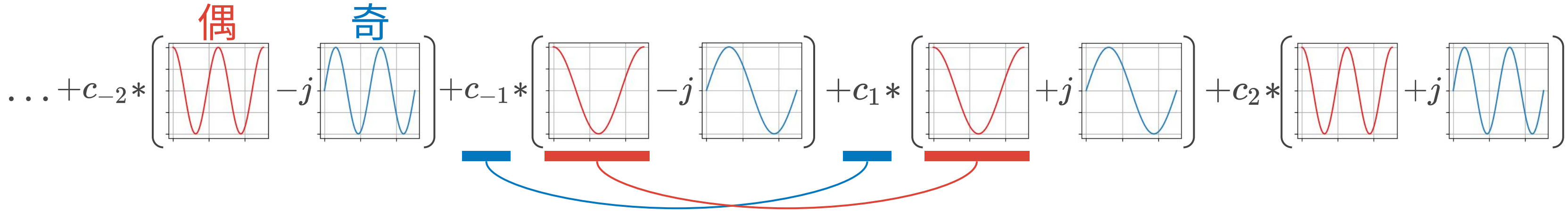
関数が**偶関数**なら係数は**実数**に，**奇関数**なら**純虚数**に．**それ以外**なら**複素数**になる．

偶



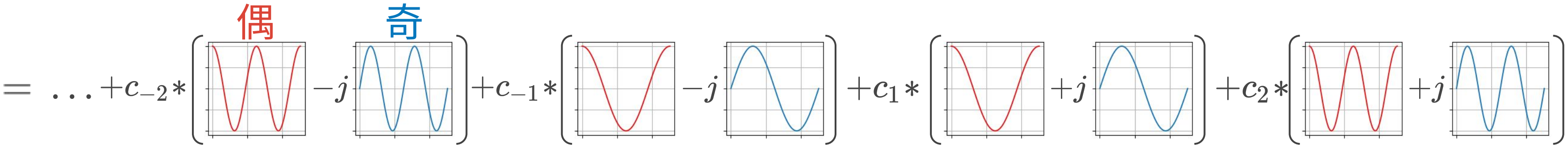
$c_{-k} = c_k$ かつ実数であればよい　←　奇関数が打ち消しあうには

奇



$-c_{-k} = c_k$ かつ純虚数であればよい　←　偶関数が打ち消しあうには

それ以外



それ以外の場合は複素数．関数が実数関数の場合は複素共役 $c_{-k} = c_k^*$

演習：前のページの内容を証明せよ．

1. 関数 x(t) が**偶関数**のとき，c_k の**虚部は常に 0** であることを示せ．

2. 関数 x(t) が**奇関数**のとき，c_k の**実部は常に 0** であることを示せ．

3. 関数 x(t) が実数関数のとき，c_k と c_{-k}は**複素共役**であることを示せ．

(ヒント)

- exp の項をオイラーの公式で分解する．
- 区間 [-a, a] で奇関数を積分すると 0 になる．
- (3) は，次式の c_k と c_{-k} の項の和が実数になるときの条件を考える．

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(-j\frac{2\pi kt}{T}\right)dt$$

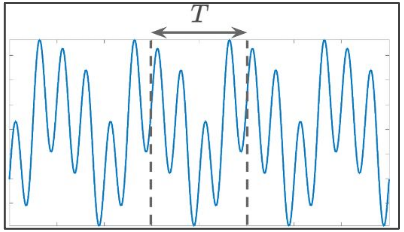
$$x(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(j\frac{2\pi kt}{T}\right)$$

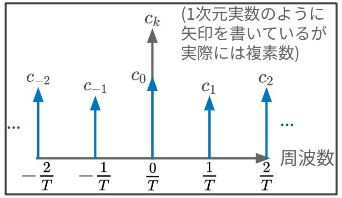
演習

手法の比較

フーリエ級数展開

周期**連続**時間信号 → **離散**周波数



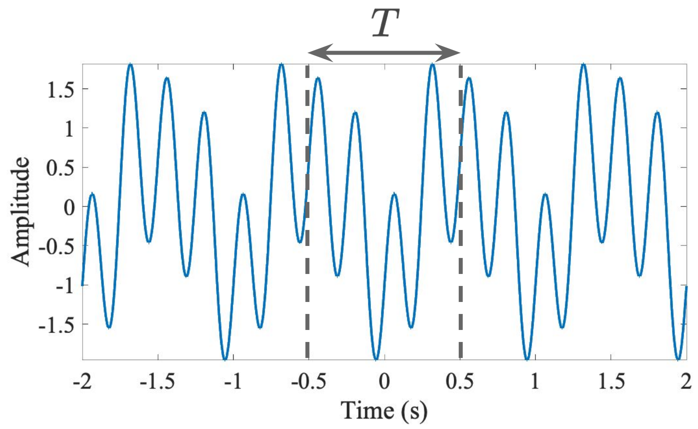


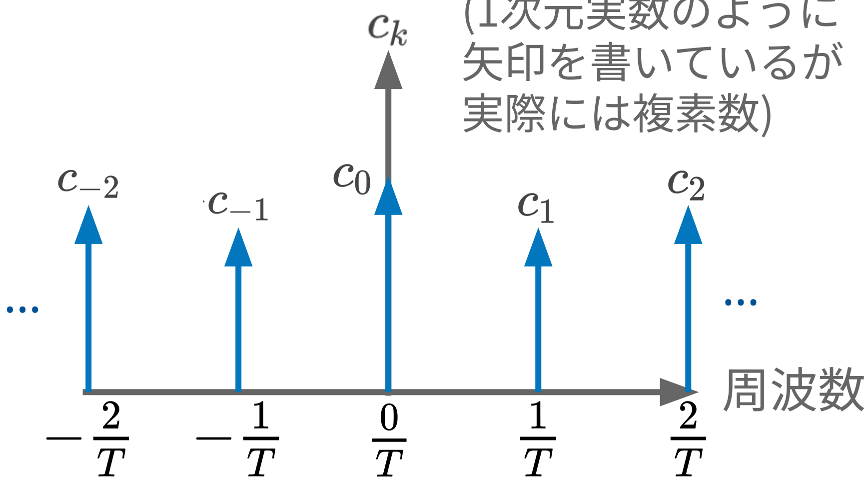
フーリエ変換
(Fourier transform)

フーリエ級数展開を図示すると

複素数　　周波数は k / T

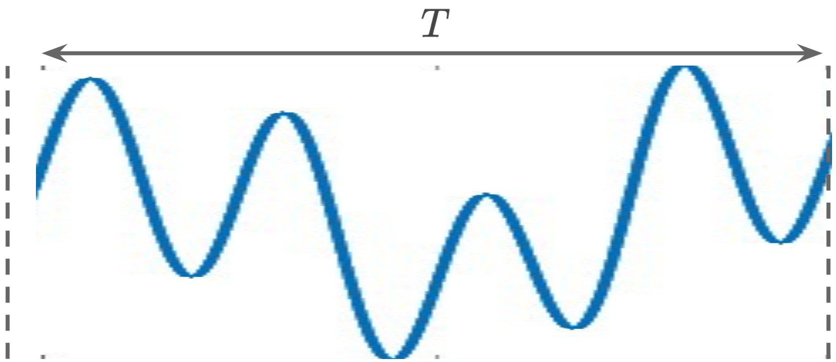
$$x(t) = \sum_{k=-\infty}^{\infty} c_k \exp\left(j\frac{2\pi k t}{T}\right) \qquad c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\exp\left(-j\frac{2\pi k t}{T}\right)dt$$

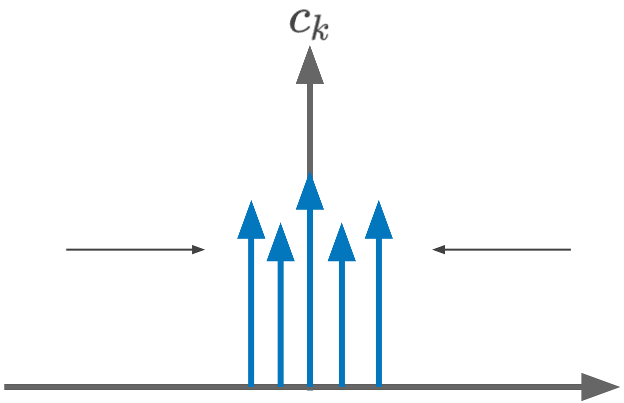




(1次元実数のように矢印を書いているが実際には複素数)

フーリエ級数展開からフーリエ変換へ：周期を無限にしてみる





周期が大きくなったら，周波数の間隔は反比例して狭くなる．

→ **周期が無限（すなわち非周期関数）になったら，**
周波数の間隔は 0 に近づく (離散的ではなく連続的) なのでは!?

フーリエ変換 とその逆変換 (ωは角周波数，2π×周波数)

- 離散 Fourier 変換： 連続時間信号 → 連続周波数

$$X(\omega) = \int_{-\infty}^{\infty} x(t) \exp(-j\omega t) dt$$

$$\mathcal{F}\left[x(t)\right] = X(\omega)$$

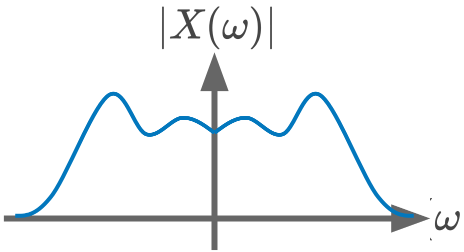
- 逆離散 Fourier 変換： 連続周波数 → 連続時間信号

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) \exp(j\omega t) d\omega$$

$$\mathcal{F}^{-1}\left[X(\omega)\right] = x(t)$$

振幅スペクトル，パワースペクトル，位相スペクトル

- $X(\omega)$ は複素数．その絶対値，絶対値の二乗，偏角に意味がある
  - $X(\omega)$ ：(角)周波数スペクトル
  - $|X(\omega)|$ ：振幅スペクトル
  - $|X(\omega)|^2$ ：パワースペクトル
  - $\arg(X(\omega))$ ：位相スペクトル



種々の性質 (詳細は応用数学の資料を参考)

- 線形性
  - “N倍はN倍，和は和のまま”

$$\mathcal{F}[ax(t) + by(t)] = aX(\omega) + bY(\omega)$$

- 相似性
  - “時間が a 倍されると周波数は1/a倍”

$$\mathcal{F}[x(at)] = \frac{1}{|a|}X(\frac{\omega}{a})$$

- 時間シフト
  - “時間シフトは周波数領域で exp 項”

$$\mathcal{F}[x(t-a)] = X(\omega)e^{-ja\omega}$$

- 周波数シフト
  - “周波数シフトは時間領域で exp 項”

$$\mathcal{F}^{-1}[X(\omega - a)] = x(t)e^{ja\omega}$$

- 変調定理
  - “余弦波の乗算は周波数の変調”

$$\mathcal{F}[x(t) * 2\cos(at)] = X(\omega - a) + X(\omega + a)$$

- 微分定理
  - “時間の微分は周波数の乗算”

$$\mathcal{F}[\frac{\partial}{\partial t}x(t)] = j\omega X(\omega)$$

- パーセバルの定理
  - “フーリエ変換はエネルギーを保存”

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} |X(\omega)|^2 d\omega$$

これらの性質は，後述する離散フーリエ変換でも成立する．

演習：前のページの内容を証明せよ．

1. フーリエ変換の線形性を証明せよ．
   a. ヒント：a x(t) + b y(t) をフーリエ変換の式に代入し，X(w) と Y(w) を作る

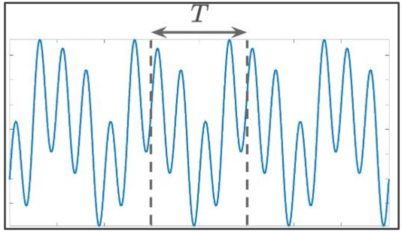
2. フーリエ変換の時間シフトを証明せよ．
   a. ヒント：同様に代入し変数変換

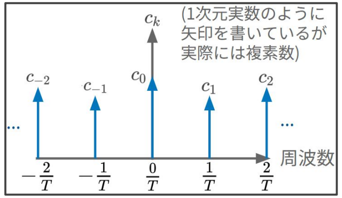
演習 (1だけ)

手法の比較

フーリエ級数展開

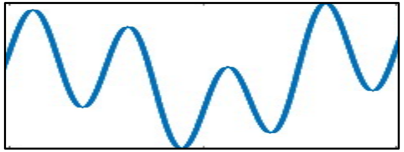
周期**連続**時間信号 → **離散**周波数

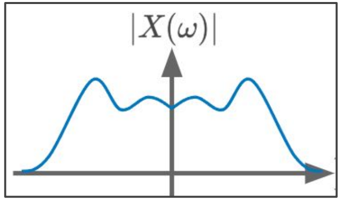




フーリエ変換

非周期**連続**時間信号 → **連続**周波数





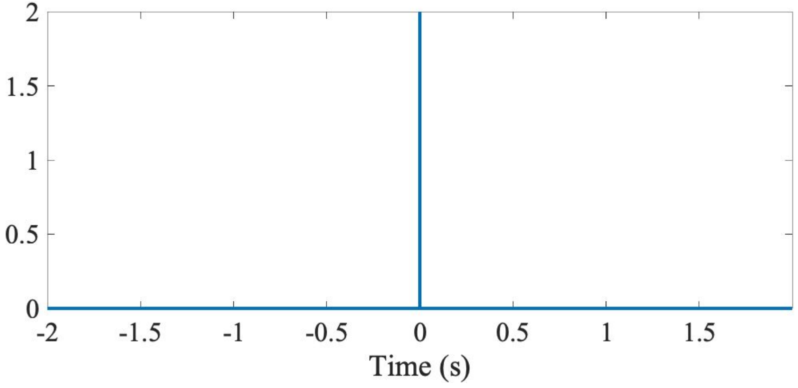
離散フーリエ変換
(Discrete Fourier transform: DFT)

復習：デルタ関数 $\delta(t)$ or $\delta[n]$

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases} \quad \delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

※厳密には正しい記述ではない．詳細は「超関数」

$\delta(t)$



$$\int_{-\infty}^{\infty} \varphi(t)\delta(t)dt = \varphi(0)$$

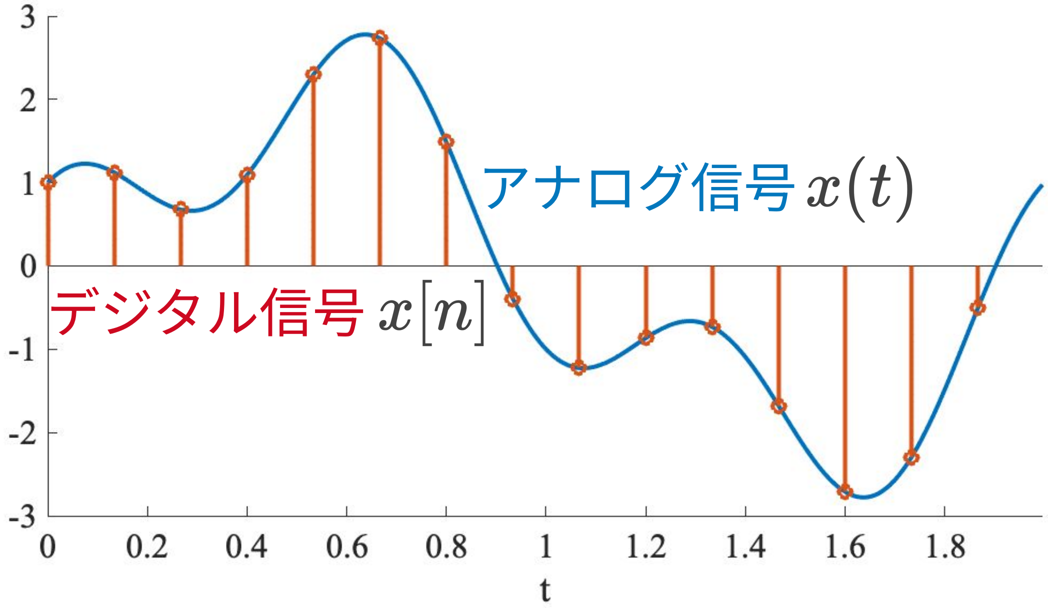
ある関数 (波形) の瞬時的な値を切り出す役割を持つ．上記の例では t = 0 の値．

復習：ディジタル信号の数式表現

$$x[n] = x(nT_s)$$

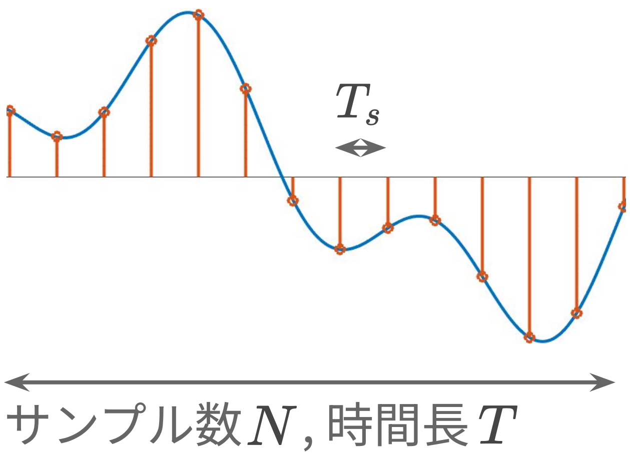
$$x[n] = x(0)\delta[n] + x(T_s)\delta[n-1] + x(2T_s)\delta[n-2] + \dots$$

離散時間信号のデルタ関数を使って書くと

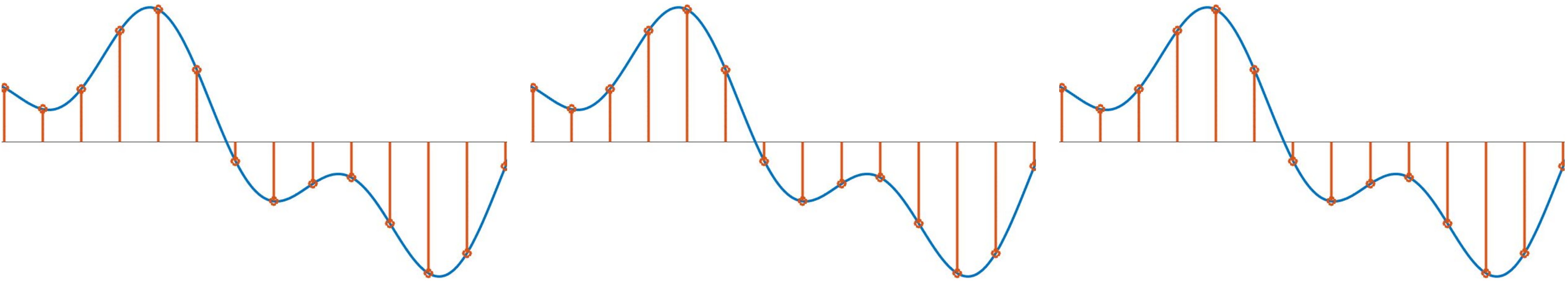


連続時間信号としての表現．有限長さの信号を考える．

$$\sum_{n=0}^{N-1} x(t)\delta(t - nT_s)$$



周期信号と見做してみる (突然ですが)



周期 T を持つ連続時間信号 → フーリエ級数展開を使ってみる

問題：デルタ信号の乗算された信号を フーリエ級数展開する

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} \underline{x(t)\boxed{\sum_{n=0}^{N-1}\delta\left(t-nT_s\right)}}\exp\left(-j\frac{2\pi kt}{T}\right)dt$$

前のページの信号

$$= \frac{1}{T}\int_{-T/2}^{T/2}\left\{x(t)\exp\left(-j\frac{2\pi kt}{T}\right)\right\}\boxed{\sum_{n=0}^{N-1}\delta\left(t-nT_s\right)}dt$$

$$\int_{-\infty}^{\infty}\varphi(t)\delta(t)dt = \varphi(0)$$

$$= \frac{1}{T}\sum_{n=0}^{N-1}\underline{x(nT_s)}\exp\left(-j\frac{2\pi knT_s}{T}\right)$$

$NT_s = T$
(サンプリング周期を N 回
繰り返したものが信号周期)

$$= \frac{1}{NT_s}\sum_{n=0}^{N-1}\underline{x[n]}\exp\left(-j\frac{2\pi kn}{N}\right)$$

$\cancel{X[k] = T_s c_k}$ と置きなおす → 離散フーリエ変換

$X[k] = NT_s c_k$

離散フーリエ変換とその逆変換

- 離散 Fourier 変換： 離散時間信号 → 離散周波数

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\frac{2\pi kn}{N}\right) \qquad \mathrm{DFT}\left[x[n]\right] = X[k]$$

$$(k \in \{0, 1, \ldots, N-1\})$$

- 逆離散 Fourier 変換： 離散周波数 → 離散時間信号

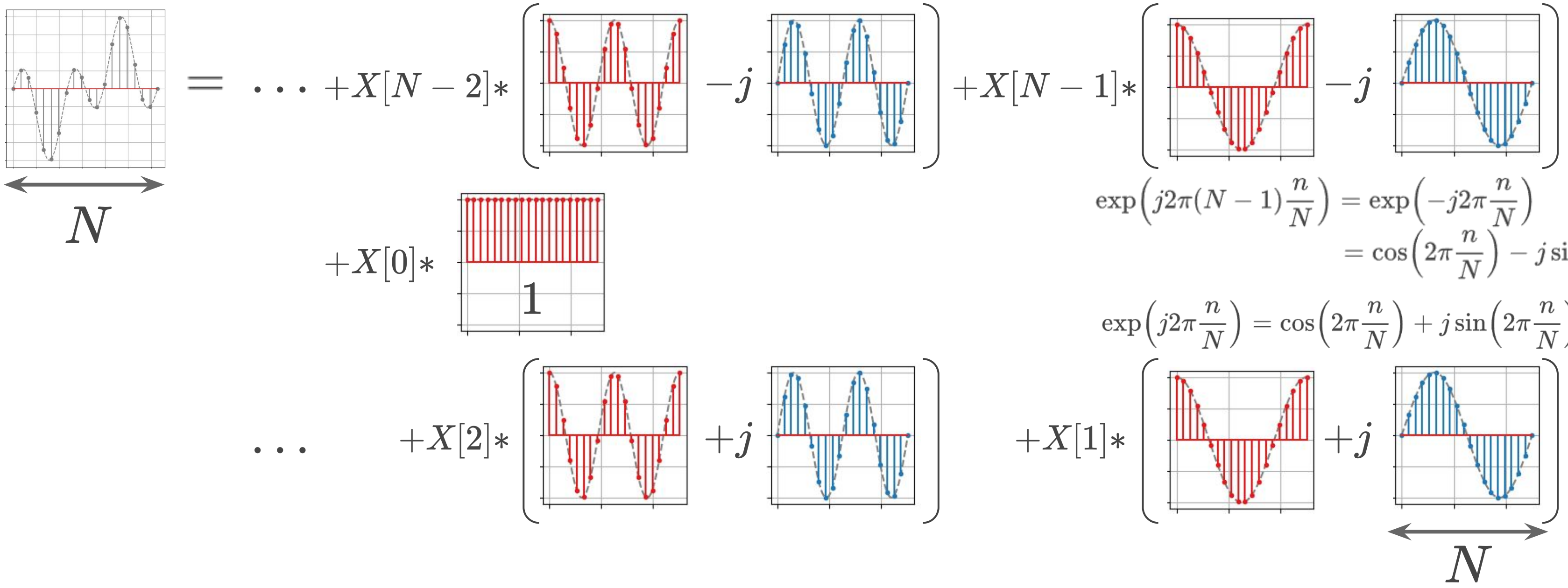
$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp\left(j\frac{2\pi kn}{N}\right) \qquad \mathrm{DFT}^{-1}\left[X[k]\right] = x[n]$$

$$(n \in \{0, 1, \ldots, N-1\})$$

その意味

複素数

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\exp\left(j\frac{2\pi kn}{N}\right)$$

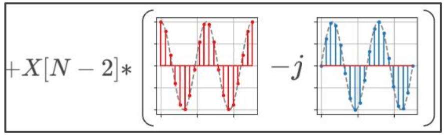


$$\exp\left(j2\pi(N-1)\frac{n}{N}\right) = \exp\left(-j2\pi\frac{n}{N}\right)$$
$$= \cos\left(2\pi\frac{n}{N}\right) - j\sin\left(2\pi\frac{n}{N}\right)$$

$$\exp\left(j2\pi\frac{n}{N}\right) = \cos\left(2\pi\frac{n}{N}\right) + j\sin\left(2\pi\frac{n}{N}\right)$$

振幅スペクトル $|X[k]|$ と位相スペクトル $\arg(X[k])$

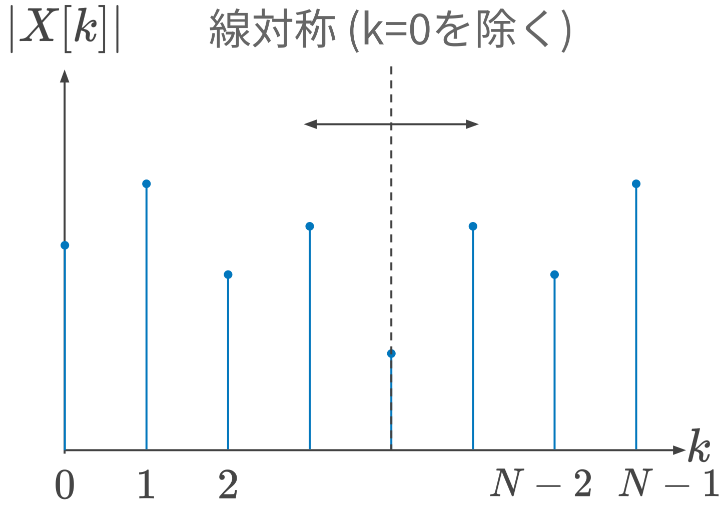
$$X[k] = \underbrace{|X[k]|}_{}\, e^{j\,\underbrace{\arg(X[k])}_{}}$$



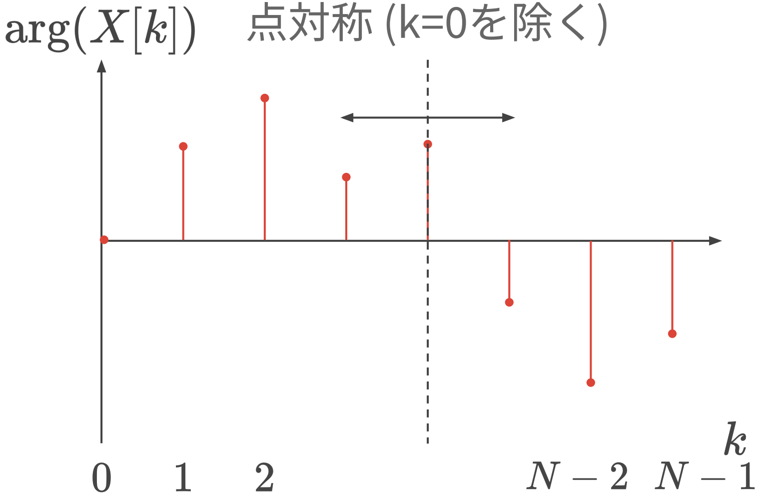
$\exp\left(j\frac{2\pi kn}{N}\right)$の大きさを何倍するか

$\exp\left(j\frac{2\pi kn}{N}\right)$の開始時刻をどれくらい動かすか

実数信号なら
線対称 (k=0を除く)



実数信号なら
点対称 (k=0を除く)



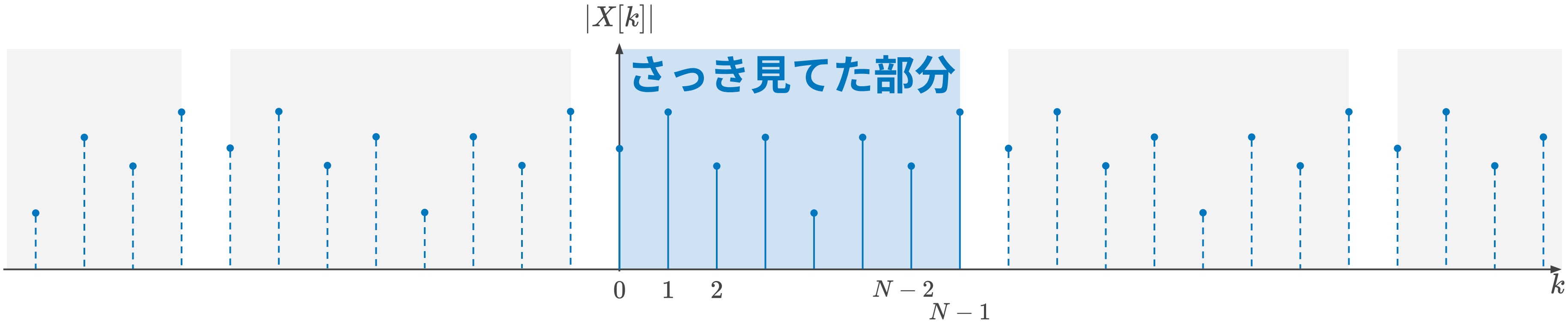
実数信号を離散フーリエ変換する場合には N-1の半分まででよい (図示する場合もそう)

$0 \leq k \leq N-1$ の外は何処に行った？

- フーリエ級数展開の k の範囲は $-\infty$ から ∞ だったのに，急に k の範囲が変わった．範囲外の k はどうなっている？実は…

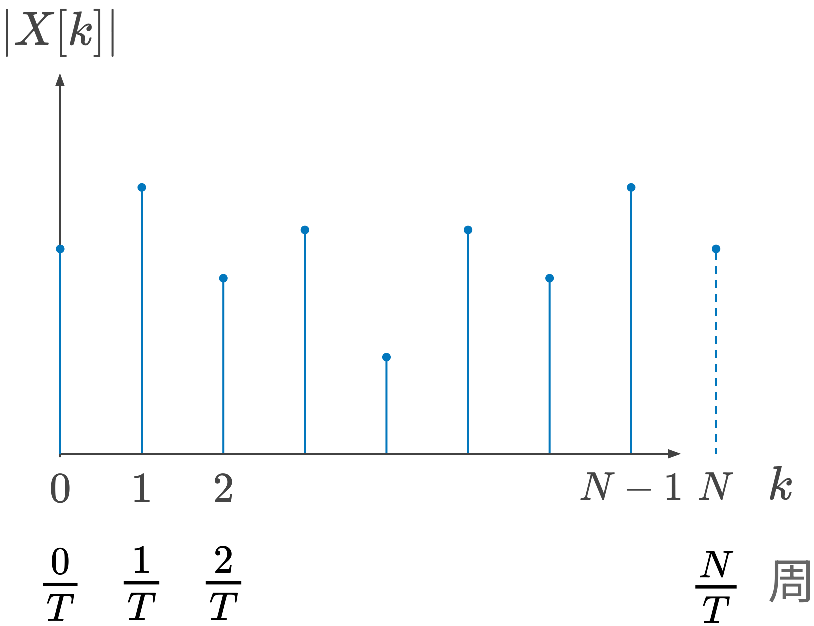
$$X[k+N] = \sum_{n=0}^{N-1} x[n] \exp\left(-j\underbrace{\frac{2\pi(k+N)n}{N}}_{=\frac{2\pi kn}{N}+2\pi n}\right) = X[k]$$

→ 周期 N で
繰り返している！



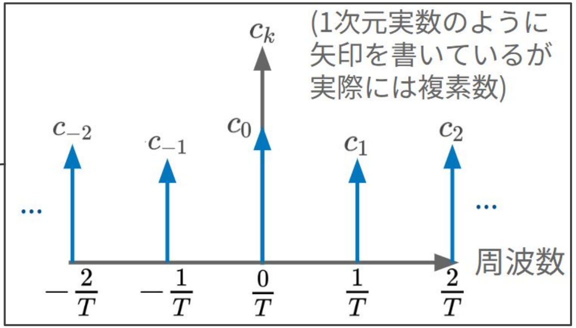
プログラムなどで扱うときは 0 ~ N-1 しか扱わないことが多いが，実はその外側では同じ周波数特性が無限に繰り返されている

サンプリング周波数との関係



$\frac{N}{T}$ 周波数

$= \frac{1}{T_s}$ (サンプリング周波数)



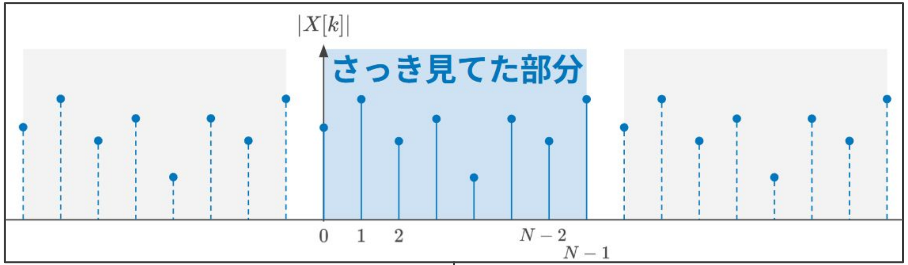
$$= \frac{1}{T}\sum_{n=0}^{N-1} \underline{x(nT_s)} \exp\left(-j\frac{2\pi knT_s}{T}\right)$$

$$= \frac{1}{NT_s}\sum_{n=0}^{N-1} \underline{x[n]} \exp\left(-j\frac{2\pi kn}{N}\right)$$

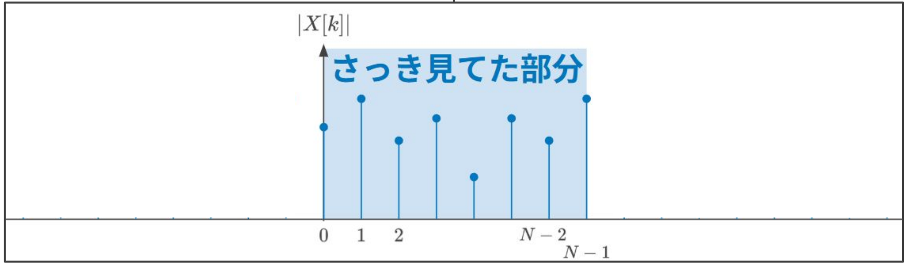
$NT_s = T$
(サンプリング周期を N 回
繰り返したものが信号周期)

k = N がサンプリング周波数に対応．すなわち，離散フーリエ変換の周波数表現は
サンプリング周波数を N 等分割して，各周波数の振幅・位相を求めている．

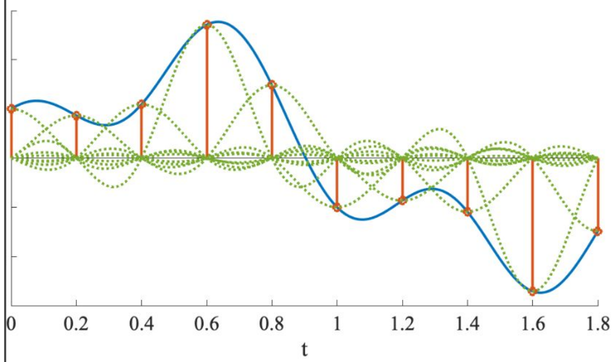
余談：ディジタル信号からアナログ変換 再訪



サンプリング周波数までを取り出す
ローパスフィルタ



等価



$$x(t) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{sinc}\left[\pi\left(\frac{t}{T} - n\right)\right]$$

ディジタル信号 → アナログ信号

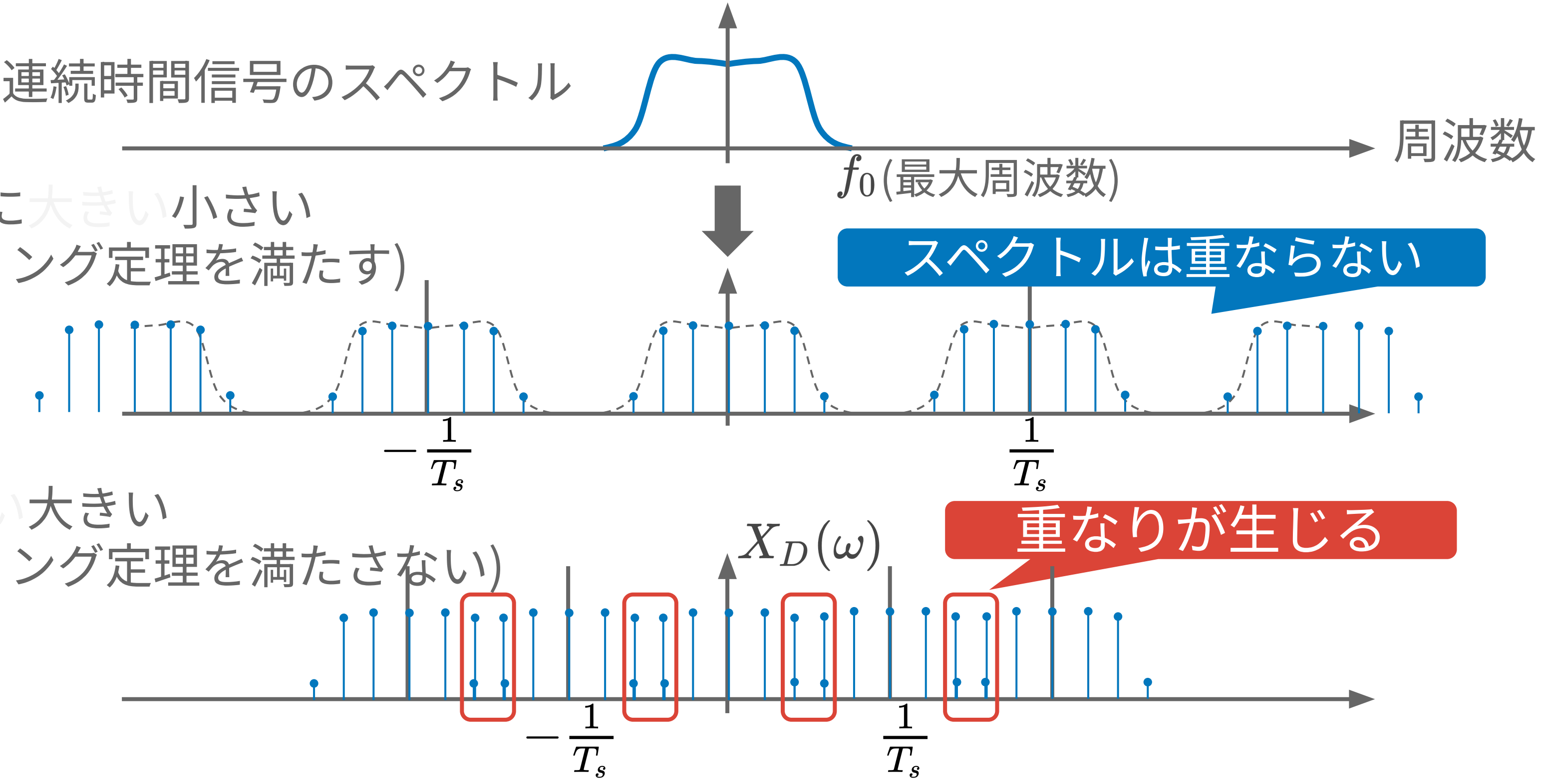
sinc 関数で畳み込み

理想ローパスフィルタ (サンプリング周波数までは 1, それ以外は 0) を X[k] に乗算することと，x[n] に sinc 関数を畳み込むことは等価．
(理想ローパスフィルタの逆フーリエ変換が sinc 関数)

サンプリング周波数定理 再訪

T_sが十分に大きい小さい
(サンプリング定理を満たす)

T_sが小さい大きい
(サンプリング定理を満たさない)



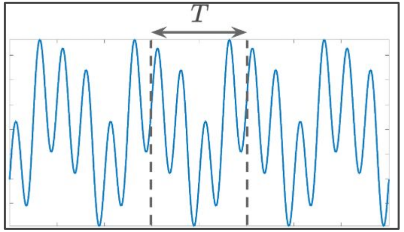
サンプリング周波数が小さいとスペクトルが重複して元スペクトルを取り出せない

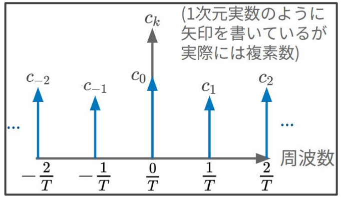
手法の比較

連続時間信号

フーリエ級数展開

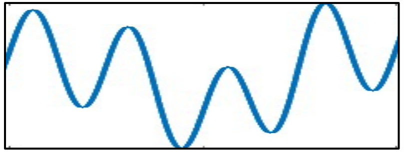
周期**連続**時間信号 → **離散**周波数

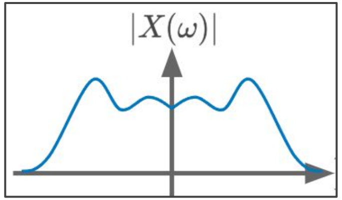




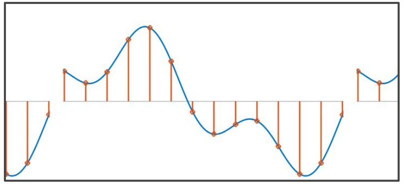
フーリエ変換

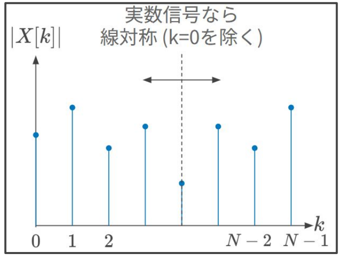
非周期**連続**時間信号 → **連続**周波数





離散時間信号





離散フーリエ変換

周期**離散**時間信号 → 周期**離散**周波数

本講義ではやりません．
興味があれば調べてください．

離散時間フーリエ変換

非周期**離散**信号 → **連続**周波数

離散周波数 (信号の周期性を仮定)　　**連続**周波数 (信号は非周期で良い)

まとめ
(matome)

まとめ

- フーリエ級数展開
  - 直交する関数系で周期連続時間信号を分解する．
  - 正弦波関数で分解する場合と，複素指数関数で分解する場合がある．
  - 信号の偶奇によって，係数の実数・純虚数・複素数が変わる．

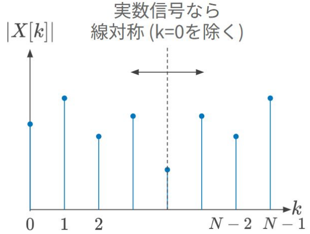
- フーリエ変換
  - 連続時間信号の周期を無限にすると，スペクトルが連続になる．
  - 離散フーリエ変換と共通する種々の性質がある．

- 離散フーリエ変換
  - 有限長で切り出した離散時間信号を対象
  - 信号も周波数スペクトルも範囲外で繰り返していると見做す

課題 (exercise)

以下の課題についてレポートを作成し，LMS 上で提出せよ。

1. 周期関数 x(t) のフーリエ級数 c_k と c_{-k}は複素共役であることを示せ



2. フーリエ変換の時間シフトを証明せよ

3. サンプリング周波数を 4000 Hz とする．離散フーリエ変換の周波数解像度 (k と k - 1 の周波数軸上の間隔) を 20 Hz としたい．そのときに，必要な信号時間長 T を答えよ

4. 離散フーリエ変換は行列として表現することができる．この行列を Python で記述せよ．詳細は次のページ

DFT の行列表現

$$\boldsymbol{X} = \boldsymbol{W}\boldsymbol{x}$$

$$\begin{bmatrix} X[0] \\ \vdots \\ X[k] \\ \vdots \\ X[N-1] \end{bmatrix} = \boxed{\begin{matrix}\text{N-by-N} \\ \text{matrix W}\end{matrix}} \begin{bmatrix} x[0] \\ \vdots \\ x[n] \\ \vdots \\ x[N-1] \end{bmatrix}$$

```
import numpy as np

def DFT_matrix(N):
  W = np.zeros((N,N), dtype=complex) # N-by-N zero matrix
  """
  Generate DFT matrix of size N.
  """
  return W

xn = np.array([1, 2, 3, 4, 5, 6, 7, 8]) # input signal
W = DFT_matrix(len(xn))

# check
print("your answer:\n", W @ xn) # Xk = W xn
print("numpy:\n", np.fft.fft(xn)) # reference. OK if your answer matches to this.
```