第4回：スペクトログラムの理解と可視化演習

音声処理と認識

担当：小野原彩香

形式：オンライン同期（Zoom）

Zoom録画を後日公開します。課題はメール/Google Driveでも提出可。

本日の到達目標

1. フーリエ変換の直感的な意味を説明できる
2. DFT・FFTをPythonで計算できる
3. STFTとスペクトログラムの仕組みを理解する
4. 窓長・FFT長・ホップ長の役割を区別できる
5. スペクトログラムから周期性・摩擦・破裂などの特徴を読み取れる

音を正弦波・余弦波の成分で表す

複雑な波形を、**振幅・周波数・位相**の異なる成分で表す。有限長の標本列はDFTで分解・再構成できる。

次は有限個の成分による模式的な表現：

$$x(t) = A_0 + \sum_{\ell=1}^{L} A_\ell \sin(2\pi f_\ell t + \phi_\ell)$$

例：ピアノの「ラ（440 Hz）」

- 基音はおよそ440 Hz。最も強い成分とは限らない
- 倍音はおよそ880 Hz、1320 Hzなど
- 実際には整数倍からのずれや打撃音も含む
- 各成分の振幅・位相と時間変化が音色に関わる

連続信号のフーリエ表現には、周期性や収束についての条件もある。

DFTの定義と周波数ビン間隔

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, \ldots, N-1$$

x[n]：N個の標本。X[k]：複素係数。|X[k]|：**未正規化**の振幅スペクトルで、元波形の振幅と同じ数値ではない。

$$f_k = \frac{k f_s}{N}, \qquad \Delta f = \frac{f_s}{N}$$

f_kは非負周波数側の対応。`np.fft.fft`の後半は負の周波数で、`np.fft.fftfreq`を使って対応させる。

- **ビン間隔と実際の分解能は別**。分解能には窓長・窓形状なども関わる
- 短い窓にゼロを補ってFFT長だけ増やしても、元の観測時間は増えない
- 直接計算のO(N^2)を、FFTではO(N log N)へ短縮する

NumPy FFTの定義・周波数・正規化

FFTをPythonで計算してみよう

```
import numpy as np
import matplotlib.pyplot as plt

fs, duration = 44100, 0.1
t = np.linspace(0, duration, int(fs * duration), endpoint=False)
x = np.sin(2*np.pi*440*t) + 0.5*np.sin(2*np.pi*880*t)

X = np.fft.fft(x)
N = len(x)
freqs = np.fft.fftfreq(N, d=1/fs)
positive = freqs > 0

plt.figure(figsize=(10, 4))
plt.plot(freqs[positive], np.abs(X[positive]))
plt.xlabel("周波数 (Hz)")
plt.ylabel("|X[k]|（未正規化）")
plt.title("FFTによる周波数スペクトル（440 Hz + 880 Hz）")
plt.xlim(0, 2000)
plt.grid(True)
plt.tight_layout()
plt.show()
```

N＝4,410点、ビン間隔は10 Hz。440 Hzと880 Hzのピークを確認する。

短時間フーリエ変換（STFT）

全区間の振幅スペクトルだけでは、変化の時刻を直接読み取りにくい。**複素DFT全体は逆変換できる**ので、時間情報を完全に失うわけではない。

$$\mathrm{STFT}\{x\}(m,k) = \sum_{n=0}^{N-1} x[n+mH]w[n]e^{-j\frac{2\pi}{N}kn}$$

m：フレーム番号、H：ホップ。ここでは窓長＝FFT長＝N。

設定	役割・例
窓長	周波数の分離と時間変化の追いやすさに関わる
FFT長	周波数ビン間隔f_s/Nを決める
ホップ	表示するフレーム中心の間隔を決める

44.1 kHzなら窓1,024点＝23.2 ms、ホップ512点＝11.6 ms（50%重複）。

式は左端基準。librosaの既定`center=True`は中心をmHへ合わせる。公式STFT

窓関数：漏れとピーク幅のトレードオフ

時間領域で窓を掛けると、周波数領域で窓のスペクトルとの畳み込みになる。端の重みを抑えることで、サイドローブを低くできる。

$$w[n] = \tfrac{1}{2}\left(1 - \cos\frac{2\pi n}{N-1}\right), \quad 0 \le n < N$$

上は**対称形**Hann。librosaの`window="hann"`は通常、分母Nの**周期形**を使う。Hammingの端は0にならない。

同じ窓長での傾向	主ローブ・サイドローブ
矩形	主ローブが狭い。サイドローブが高い
Hann・Hamming	端の重みを抑え、主ローブ幅と漏れのバランスを取る
Blackman	サイドローブを抑えるが、主ローブは広がる

ビンに一致する純音なら、矩形窓でも他のDFTビンへの漏れは出ない。SciPyのHann窓

スペクトログラム：周波数軸と色の尺度を分ける

表示する量	設定
横軸	時間（秒）
縦軸	周波数：線形Hz、対数周波数、メルなど
色	振幅・パワーの線形値、または基準値に対するdB

$$L_P = 10 \log_{10} \frac{P}{P_{\text{ref}}}, \qquad L_A = 20 \log_{10} \frac{A}{A_{\text{ref}}}$$

- 低域を広く表示するのは**周波数軸**の設定
- 強弱の広い範囲を表示するのは**色のdB**の設定
- `amplitude_to_db(abs(D), ref=np.max)` は最大値を0 dBとする。既定では80 dB下で切り詰める

この相対dBは音圧レベル（dB SPL）ではない。別々に最大値正規化した図から絶対的な音量差は読めない。

有声音・無声音・非発話区間の見え方

種類	例	スペクトログラムの手がかり
有声音	母音の多く、/m/・/n/	狭帯域では調波。広帯域では周期的な縦縞も
無声音	子音/s/・[ɕ]、/t/・/k/の閉鎖・破裂	摩擦は雑音状、破裂は短いバースト、閉鎖は低エネルギー
声がない区間	発話の間など	環境雑音が残る場合もある

「さ」全体は子音と母音を含むので、一律に無声音としない。色の暗さだけで無声・非発話を区別できない。

- 有声音のF0は、準周期的な声帯振動の繰り返し周波数
- 十分長い窓・線形Hz軸では、調波がほぼF0間隔に並ぶ
- F0が動けば縞も曲がる。/s/の高域成分も話者・調音・録音帯域で変わる

日本語母音のフォルマント：実習用の設定例

フォルマントは**スペクトル包絡の共鳴の山**。個々の調波やF0とは区別する。

母音	F1（Hz）	F2（Hz）	調音の目安
あ /a/	800	1200	大きく開く
い /i/	300	2500	前舌で狭める
う /u/	400	1400	唇を強く突き出さない。標準語では[ɯ]など
え /e/	500	1900	開口は中間程度
お /o/	500	800	唇を丸める

値は**合成母音の模式的な設定**。日本語話者の平均値・判定基準ではない。

- F1はおおむね開口が大きいほど高く、F2は前舌で高い傾向
- 声道形状との一対一の対応ではなく、話者・文脈で変わる

東京外国語大学：日本語の「う」

演習1：スペクトログラムの描画（Colab）

```
import librosa, librosa.display
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Audio, display

# LibriSpeechの英語朗読音声。初回はダウンロード
y, sr = librosa.load(librosa.example("libri1"), sr=None)
display(Audio(y, rate=sr))

n_fft, hop_length = 2048, 512
D = librosa.stft(y, n_fft=n_fft, hop_length=hop_length,
                 window="hann")
S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
fig, ax = plt.subplots(figsize=(12, 4))
img = librosa.display.specshow(
    S_db, sr=sr, hop_length=hop_length, x_axis="time",
    y_axis="log", ax=ax, cmap="magma")
ax.set_title("スペクトログラム（対数周波数軸・相対dB）")
fig.colorbar(img, ax=ax, format="%+.0f dB")
plt.tight_layout()
plt.show()
```

窓時間＝n_fft/sr。上の時間設定は、理論の25 ms窓の例とは異なる。

演習2：調波の強調で母音の違いを表す

本文のPython実習・セル4で`make_vowel`を定義してから、次を実行する。共鳴の包絡Eで、F0の整数倍の成分を重み付けする。

$$x(t) = \sum_{\substack{k \geq 1 \\ k f_0 < f_s/2}} \frac{E(k f_0)}{k} \sin(2\pi k f_0 t)$$

```
vowels = {"あ": [800,1200,2600], "い": [300,2500,3200],
          "う": [400,1400,2400], "え": [500,1900,2700],
          "お": [500,800,2400]}
f0, sr, n_fft, hop = 150, 22050, 1024, 256
fig, axes = plt.subplots(1, 5, figsize=(18, 4))
for ax, (name, fmts) in zip(axes, vowels.items()):
    sig = make_vowel(f0, fmts, sr=sr)
    D = librosa.stft(sig, n_fft=n_fft, hop_length=hop)
    S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)
    librosa.display.specshow(S_db, sr=sr, hop_length=hop,
        x_axis="time", y_axis="hz", ax=ax, cmap="magma")
    ax.set_title(name); ax.set_ylim(0, 5000)
plt.tight_layout(); plt.show()
```

自然音声を完全に再現するモデルではない。全母音でF0＝150 Hz、各音・各図を個別にピーク正規化する。

演習3：ブラウザ録音を変換して可視化する

本文のセル5で`record_js`を定義する。これは3秒録音の完了後に解決するPromiseを返す。

```
from google.colab import output
print("マイクの許可後、3秒間話してください")
audio_data = output.eval_js(record_js)
```

続くセル6の処理	目的
データURLのbase64をバイトへ戻す	ブラウザの録音データを保存
FFmpegで内容を自動判別して変換	22,050 Hz・モノラルのPCM WAVへ
librosaで読み込み、STFTを計算	波形・スペクトログラム・メル表示を比較

録音形式をWAVやWebMに固定して解釈しない。JavaScriptは実際のMIME形式を使い、`finally`でマイクを停止する。

マイクを使えない場合はWAVをアップロードし、`y, sr = librosa.load("ファイル名.wav", sr=None)`から分析を実行できる。

演習課題（授業中）

課題1：発話サンプルを読み取る

1. `libri1` の音声を聞き、有声・無声の区間候補を時刻で示す
2. 十分長い窓・線形Hz軸で、調波の間隔からF0を目視推定する
3. 同じ窓長で `window="boxcar"` と `window="hann"` を比較する

課題2（任意）：自分の声で比較

1. 「あ」と「い」を約1秒ずつ発声して録音する
2. F1・F2の付近をスペクトル包絡の山として観察する
3. スクリーンショットに矢印を加えて共有する

個々の調波とフォルマントを混同しない。表示条件も結果と一緒に記録する。

次回（第5回）の予告

形式：動画回（非同期）

テーマ：MFCC

> 従来の音声認識で広く使われてきた特徴量を学ぶ。
> 対数メルスペクトログラムとの違いも確認する。

予習課題（やれる範囲で）

- 「メル尺度」「ケプストラム」を調べる
- 第5回の動画教材を視聴してからテストに取り組む

第4回のまとめ

概念	ポイント
DFT・FFT	正弦波・余弦波の成分へ分解し、FFTで効率よく計算
STFT	窓を掛けた短い区間ごとにDFTし、時間変化を表示
窓とFFT長	実際の分解能と、FFTビンの刻みを区別
窓関数	サイドローブの漏れを抑えるが、主ローブは広がる
スペクトログラム	時間・周波数・色の強度。周波数軸とdB色は別
有声音・無声音	周期性、摩擦、閉鎖、破裂などの手がかりを読む
フォルマント	共鳴の包絡の山。F0や個々の調波とは別

提出：演習のスクリーンショットを翌週末23:59までにGoogle Driveまたはメールで提出。