離散 Fourier 変換を試してみる

桂田 祐史

2015 年 11 月 24 日

1 まずやってみよう

1.1 準備

(1) この授業の WWW サイト[1] から、ギターのドの音の WAVE ファイル `guitar-5-3.wav` を入手し、どこか適当なところ (デスクトップとかホームディレクトリィとか) に保存する。
(このファイルを開くと音が鳴るが、今日の授業中は気楽にやってよし。)

(2) Mathematica を起動して、カレント・ディレクトリィを (1) のディレクトリィにする (ホームディレクトリィに保存した場合は必要がない)。

具体的なコマンドは、例えばデスクトップに保存した場合は

```
SetDirectory["~/Desktop"]
```

`FileNames[]` というコマンドを入力すると、ファイル名一覧が見える。

1.2 guitar-5-3.wav の音を離散 Fourier 変換する

以下の実験の内容は、おおむね古い卒研のレポートに基づく (松山 [1])。

```
snd=Import["guitar-5-3.wav"]
```

これでファイル `guitar-5-3.wav` を変数 `snd` に読み込む。ボタンを押すと音が再生できる。

```
tbl = snd[[1, 1, 1]];
```

左チャンネルの音の数値データを `tbl` に代入した。

右チャンネルの音が欲しければ `tbl=snd[[1,1,2]]` とするか、あるいは `{ltbl,rtbl}=snd[[1,1]]` として同時に代入する。ちなみに `snd[[1,2]]` はサンプリング周波数である。

```
sr=snd[[1,2]]
```

44100 となるはずである。音楽用 CD と同じ、44.1 kHz というサンプリング・レートで録音したことを示している。

[1] http://nalab.mind.meiji.ac.jp/~mk/fourier/

```
tb = Take[tbl, {1, 3*sr}];
g = ListPlot[tb, PlotRange -> All]
```

3 秒分取り出してグラフをプロットしてみた。

(サンプリング周波数が $\mathrm{sr} = 44.1$ kHz なので、1 から $3 * \mathrm{sr}$ で、3 秒分のデータということになる。`Take[]` はリストからデータを取り出す関数である。)

```
tb = Take[tbl, {62800+1, 62800+sr}];
g = ListPlot[tb, Joined -> True, PlotRange -> {{1, 1600}, {-0.3, 0.3}}]
```

音が鳴り始めるのは 62800 番目辺りからなので、そこから 1 秒分 ($\mathrm{sr} = 44100$ 個のデータを) 取り出して、1600 個分 ($1600/44100 \fallingdotseq 0.036$ 秒分) プロットしてみた。ここは色々試してみると良い。

```
ListPlay[tb, SampleRate->sr]
```

とすると、取り出したデータ `tb` の音を鳴らすことが出来る。

```
c = Fourier[tb];

ListPlot[Abs[c], Joined->True, PlotRange->All]
```

`tb` の離散 Fourier 変換 `c` を求め、絶対値をプロットした。これから周波数の分布が読み取れる (はず)。

`Abs[]` の代わりに `Re[]`, `Im[]` としてみたり (やや分かりにくいけれど、$C_n = \overline{C_{N-n}}$ の関係が見える？)。

```
graph[c, n1_, n2_] := ListPlot[Abs[c], Joined -> True,
        PlotRange -> {{n1, n2}, {0, Max[Abs[c]]}}]

graph[c, 1, 1600]
graph[c, 250, 270]
```

範囲を区切って表示することで、ピークを探してみた。ピークは 130 番目である。つまり $|C_{129}|$ が最大ということである。(リスト `c` の 1 番目の要素は C_0 であるので、リストの要素の番号と Fourier 係数のインデックスが 1 ずれていることに注意する。)

これは実はこのバイオリンの音の基本周波数が 129 Hz (ドの周波数 131 に近い — ぴったりでないのは、チューニングを少し失敗している) であることを意味する。(この辺は次回説明する予定である。)

手近の楽器の音 (ピアノ、リコーダー、…) を自分で録音出来る人は、やっておいて下さい。

参考文献

[1] 松山周五郎：音の Fourier 解析, 2003 年度卒業研究レポート, http://nalab.mind.meiji.ac.jp/~mk/labo/report/open/2003-matsuyama.pdf (2004).

2　PCM による音のデジタル信号表現

音とは空気中を伝播する縦波である。音があるとき、気圧が音がないときの圧力を基準に時間変化する。圧力の基準からの変化を音圧と呼ぶ。音圧の時間変化を記録することで音が記録出来る。

PCM (pulse code modulation, パルス符号変調) とは、アナログ信号 (連続変数の関数) をデジタル信号 (離散変数の関数 — 数列) で表現するための 1 つの方法であり、音楽用 CD, コンピューターのデジタル・オーディオ, デジタル電話等で標準的な形式となっている。具体的には、次の二つに基づく。

(a) 一定の時間間隔で信号の値を評価する (サンプリングする、という)

(b) 信号の値は有限桁の数で表現する (量子化する、という)
特に値の属する区間を等間隔に区切って、もっとも近い値に丸めることで実現するとき、LPCM (linear PCM) という。

パソコンで音を記録 (録音) する場合は、電気的な信号に変換された音声を数値化して (アナログ信号からデジタル信号に変換するので **AD 変換** (analog-to-digital conversion) という)、数値を記録することになる。

サンプリング周波数とは、1 秒間に何回サンプリングするかを意味している。サンプリング周波数が高いほど、より高い音が記録できることになる。

音楽用 CD (1980 年に SONY と Phillips により規格化された) では、サンプリング周波数 44.1 kHz が採用された。

サンプリング周波数が 44.1kHz とは、1 秒間に 44100 回のデータを記録するということになる。これが採用された理由は主に次の理由による。

(a) 人間が普通聞くことが出来る音の周波数は 20 ～ 20kHz と言われている。

(b) サンプリング周波数は、再生したい最も高い音の周波数の 2 倍以上にする必要がある (これは後で解説する**サンプリング定理**を根拠としている)。

コンピューターで処理することを考えると、「有限桁の数」は、「2 進法の有限桁の数」ということになるが、その際の桁数 (ビット数) を**量子化ビット数**と呼ぶ。

8 ビットの場合は $2^8 = 256$ 段階、16 ビットの場合は $2^{16} = 65536$ 段階で表現することになる。

サンプリング周波数と量子化ビット数の値が大きいほど、元の信号により忠実なデータが得られるが、データの量はそれだけ増大する。

3

(これまで独立変数は x と書くことが多かったが) 音声信号を考えるとき、独立変数は時刻なので、ここでは t と書くことにする。

信号の値そのものは x で書いてある、つまり信号を $x(t)$ としている本が多いので、ここでもそれに従う。

周期 T の周期関数 $x\colon \mathbf{R}\to\mathbf{C}$ は次のように Fourier 級数展開出来る。

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi t}{T}},$$

$$c_n = \frac{1}{T}\int_0^T x(t) e^{-i\frac{2\pi t}{T}}.$$

基音の周波数は周期の逆数 $f=\dfrac{1}{T}$ である。n 倍音の周波数 n_0/T に対応するのは $n=\pm n_0$ の項である。

$[0,T]$ の N 等分点での値を用いる。サンプリング周期 $T_s=T/N$, サンプリング周波数 $f_s=\frac{N}{T}$ でサンプリングする、ということになる。

$[0,T]$ を N 等分点 $t_j=jT_s$ での x の値 $x_j=x(t_j)$ を用いる。

$$C_n=\frac{1}{N}\sum_{j=0}^{N-1}x_j\omega^{-nj},\quad \omega=e^{2\pi i/N}.$$

4 Mathematica での音の取り扱い

実は Mathematica の Fourier[] がデフォルトで計算するのは

$$C_n'=\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}x_j\omega^{nj}$$

である。C_n を計算させるには、`FourierParameters->{-1,-1}` というオプションを与えれば良い。ところで、信号が実数値であることから、$C_n'=\sqrt{NC_n}$ という関係にある。ゆえにパワースペクトルについては、$|C_n|^2=\frac{1}{N}|C_n'|^2$ であるから、周波数を求めたりする場合は (パワースペクトルが大きくなる n はどこか調べる)、デフォルトのまま使っても良い。

最近のコンピューターを用いて音データを取り扱うための情報については、(かなり雑ではあるが)「音の取り扱いに関するメモ」[2] が参考になると思われる。

`Import[]`, `ListPlay[]`, `Fourier[]`

[2] `http://nalab.mind.meiji.ac.jp/~mk/labo/text/memo-sound/`