

[Scilab で信号処理 2010]

情報工学科 教授 幹 康
ymiki@cs.takushoku-u.ac.jp

©2010 Yasushi Miki

[1] 時報の音はどんな音?

[以下に示す、[青色のプログラム](#)は Scilab のコンソール上で実行可能です。]

■ Scilab の起動

Scilab を起動するとコンソール画面 (Scilab console) が開かれます。このコンソール上でプログラムを入力、実行させることができます。グラフィックスはグラフィック・ウィンドウが開かれて描かれます。

■ グラフィック・ウィンドウの初期設定

グラフィック・ウィンドウの位置、サイズなどを設定します。

`gdf()` : デフォルト・グラフィック・ウィンドウ情報を取得。(get default figure)

`gda()` : デフォルト座標軸情報を取得。(get default axes)

```
f = gdf(); a = gda();  
f.figure_position = [800,40];  
f.figure_size = [480,320];  
f.axes_size = [476,226];  
a.x_location = "origin";
```

これらの設定は、Scilab のメニュー「アプリケーション(A)」の中にあるエディタ (SciNotes) を利用して入力し、'DefaultWindow.sce' のようなファイル名で保存しておけば、Scilab起動時に実行できて便利です。

```
exec('DefaultWindow.sce');
```

他に

`clf()` : グラフィックスの消去。

`scf()` : 新規ウィンドウを開く。(set current figure)

`gcf()` : カレント・グラフィック・ウィンドウ情報を取得。(get current figure)

などの関数が用意されています。

■ 時報の音

普段、テレビやラジオで聴いている時報はどんな音だろうか？

最近のパソコンはマイク入力を有するものが多く、簡単に音を収録できます。

「ポッ・ポッ・ポッ・ポーン」の最後の長い音を録音してファイルにしたのが次の "TimeSignal.wav" です。

サンプリング周波数は 44,100 Hz です。

■ サウンドファイルのダウンロード

この文書と同様にサウンドファイル "TimeSignal.wav" をダウンロードしてください。

■ Scilab へのサウンドファイルの読み込み

```
cd
```

と入力すると、現在のディレクトリを自分の home に変更することができます。

サウンドファイルを読み込むには、wavread() という関数を使います。また、ダウンロードしたファイルのディレクトリをカレントディレクトリに設定する必要があります。

```
cd /Users/                // サウンドファイルのあるディレクト  
リを Users に設定します。
```

```
x = wavread('TimeSignal.wav');
```

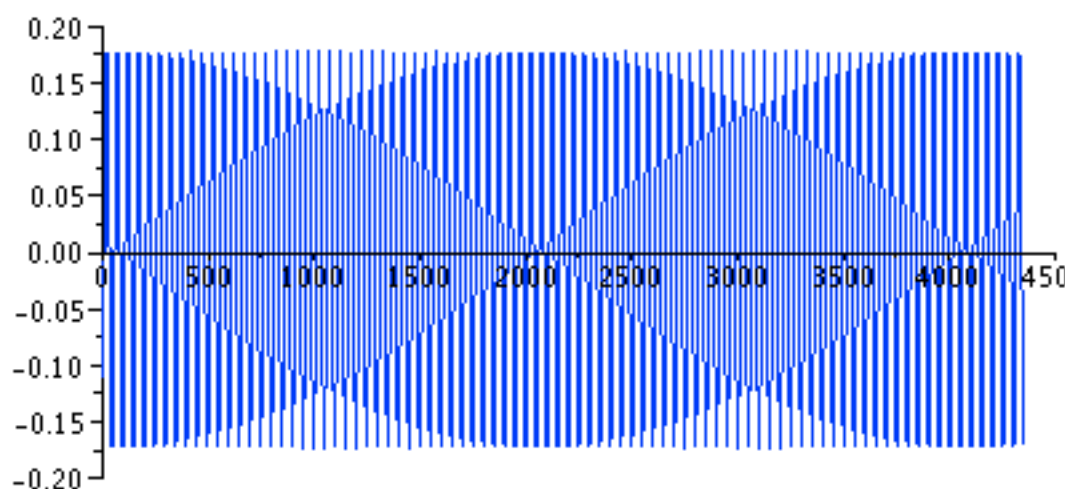
これで、配列 x にデータが読み込まれます。Scilab では前もって変数、配列などを宣言しておく必要はありません。行末のセミコロン (;) は評価された値の出力を抑制します。読み込んだデータの長さは

```
length(x)  
ans =
```

```
4352.  
で確認できます。
```

続いて、波形を表示してみましょう。

```
scf();  
plot(x);
```

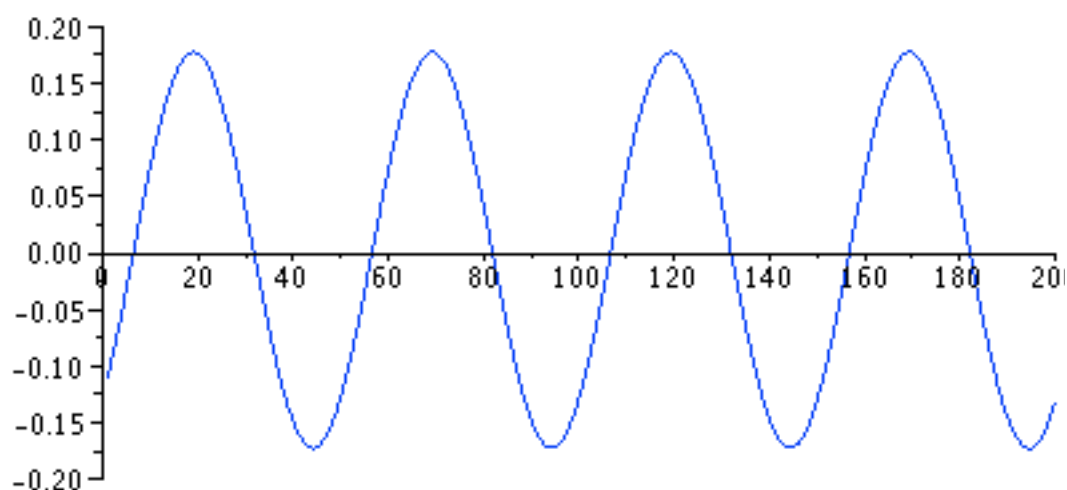


Window 版の Scilab の場合は関数 `playsnd()` を用いることによって音を鳴らすことができます。Macintosh では対応していないようです。「iTune」などを利用して、別途確認しましょう。

```
playsnd(x, 44100);
```

一部分だけ拡大して表示するには、配列 `x` の範囲を `x(1:200)` のように指定します。

```
scf();
plot(x(1:200));
```



ほとんど正弦波であることが分かります。

この波形から、信号の周期を概算することができます。

200点のデータの中に、およそ、4周期入っていることがわかります。すなわち、1周期は50点です。これにサンプリング周期を掛けてやれば、信号の周期が求まります。

サンプリング周期 DT はサンプリング周波数の逆数ですから、

$$DT = 1 / 44100$$

DT =

$$0.0000227$$

[s] したがって、信号の周期 T は

$$T = 50 * DT$$

T =

$$0.0011338$$

約 1.1 ms となります。

この逆数をとって周波数に変換すると

$$F = 1/T$$

F =

$$882.$$

約 880 Hz となります。

以上は、時間波形から概算した結果です。

■ フーリエ変換によるスペクトル分析

では、フーリエ変換を計算し、信号のスペクトルを求めてみましょう。フーリエ変換を計算するプログラムは、通常、FFT（高速フーリエ変換）が利用されます。FFTを実行するためにデータ点数を調整します。点数は2のべき乗が用いられますから、この場合は $2^{12} = 4096$ にします。データ点数の変更は次のようにすれば良いです。

$$y=x(1:4096);$$

このサウンドの長さは時間に換算すると

$$4096 * DT$$

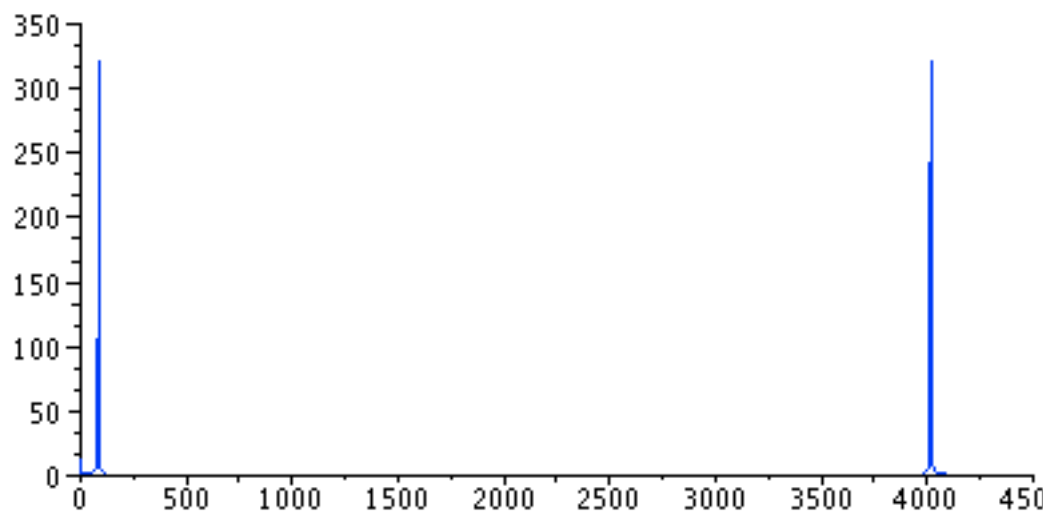
ans =

$$0.0928798$$

約 93 ms です。

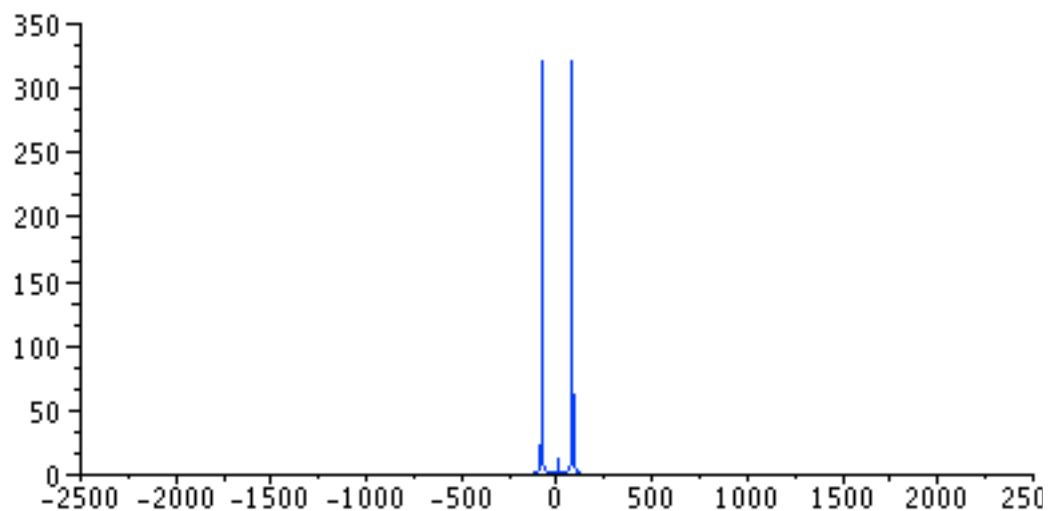
Scilab には FFT の関数 `fft()` が用意されています。フーリエ変換は一般に複素数ですから、グラフにするには絶対値などで表示させる必要があります。絶対値関数として `abs()` が使用できます。

```
Y = fft(y);  
scf();  
plot(abs(Y))
```



右端の方に現れている成分は負の周波数成分に対応しています。右半分を左側に接続するとわかりやすいかも知れません。

```
plot2d(-2047:2048,[abs(Y(2049:4096)),abs(Y(1:2048))],style = 2);
```

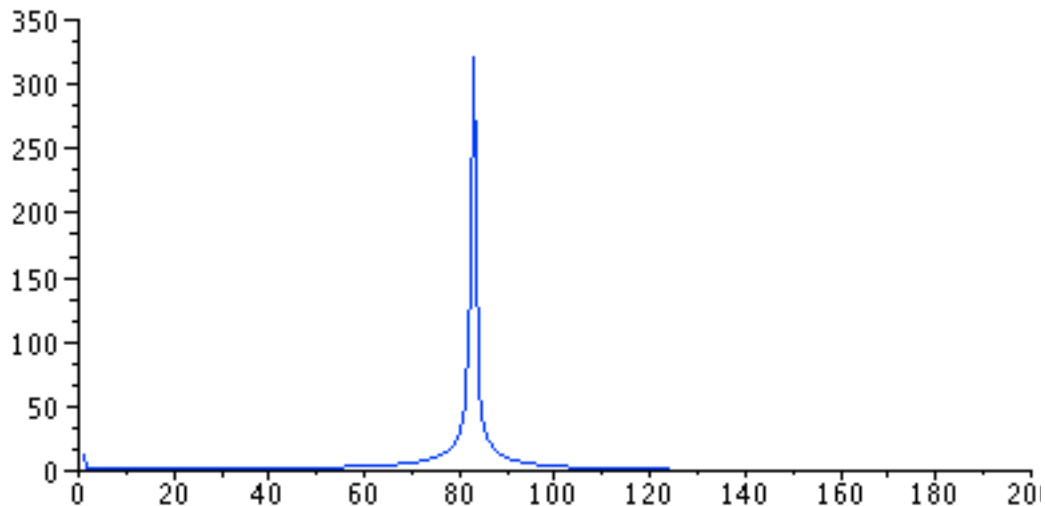


1 つ前のグラフに戻って、スペクトルは正の周波数（左半分）と負の周波数（右半分）として対称に現れます。

スペクトル分析を行う場合、通常は、左半分に注目すれば十分です。

細かくて解りづらいので横軸の一部分を拡大しましょう。

```
plot(abs(Y(1:200)));
```



鋭いラインスペクトルになっています。信号波形を見てもほとんど正弦波だったので、納得できます。

では、このピークの周波数はいくつでしょうか。絶対値が 100 を超える配列番号を表示すると

```
for i=1:100,  
    if abs(Y(i)) > 200 then printf("i = %d\n",i)  
    end  
end
```

i = 83

i = 83がピークです。

この配列番号に対応する周波数を求めるには以下の情報が必要です。

データのサンプリング周波数：f = 44,100 [Hz]

フーリエ変換の点数：N = 4,096

スペクトルの周波数間隔（周波数刻み）はサンプリング周波数を点数で割ったものとなるから

$Df = 44100 / 4096$

Df =

10.7666

Scilab の配列添字は1から始まります。つまり、直流成分（周波数 0）が添字 1 で与えられ、添字 2 の周波数がスペクトル分析の最小単位（周波数刻み Df ）となります。従って、 i 番目に対応する周波数は 1 を減じて、 $f = (i - 1) \times Df$ となることに注意してください。では、ピーク周波数は

$f = (83 - 1) * Df$
f =

882.86133

単位はもちろん Hz です。およそ、880 Hz の「ラ」の音であることがわかります。

このような周波数分析では、周波数刻みより細かい分析はできません。推定精度は周波数刻みで決まってしまうます。この場合、 $\pm Df / 2 = \pm 5$ Hz の誤差が含まれる可能性があります。