情報解析学（2023 年度後期）

担当：大阿久 俊則

第 12 回（12 月 6 日）

- 2.2 逆離散フーリエ変換と標本化定理（続き）
- 2.3 MATLAB による音声解析

2.2 逆離散フーリエ変換と標本化定理

定理 2.1（逆離散フーリエ変換）

$(\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$ を $(f_0, f_1, \ldots, f_{N-1})$ の離散フーリエ変換とするとき，

$$f_k = \sum_{n=0}^{N-1} W^{nk}\, \tilde{c}_n \qquad (k = 0, 1, \ldots, N-1) \tag{1}$$

が成立する．

(1) を行列で表せば，

$$\begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & \cdots & W^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & W^{2(N-1)} & \cdots & W^{(N-1)^2} \end{pmatrix} \begin{pmatrix} \tilde{c}_0 \\ \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_{N-1} \end{pmatrix} \tag{2}$$

となります．これを**逆離散フーリエ変換**といいます．

例 **2.3** $N=4$ のとき，$W=\exp(\frac{\pi i}{2})=i$ より離散フーリエ変換と逆離散フーリエ変換は次のようになります.

$$\begin{pmatrix}\tilde{c}_0\\ \tilde{c}_1\\ \tilde{c}_2\\ \tilde{c}_3\end{pmatrix}=\frac{1}{4}\begin{pmatrix}1&1&1&1\\1&-i&-1&i\\1&-1&1&-1\\1&i&-1&-i\end{pmatrix}\begin{pmatrix}f_0\\ f_1\\ f_2\\ f_3\end{pmatrix}$$

$$\begin{pmatrix}f_0\\ f_1\\ f_2\\ f_3\end{pmatrix}=\begin{pmatrix}1&1&1&1\\1&i&-1&-i\\1&-1&1&-1\\1&-i&-1&i\end{pmatrix}\begin{pmatrix}\tilde{c}_0\\ \tilde{c}_1\\ \tilde{c}_2\\ \tilde{c}_3\end{pmatrix}$$

逆離散フーリエ変換の式 (1) の意味を考えてみましょう．(1) を角周波数 $\omega = \dfrac{2\pi}{T}$ と標本点（分点）$t_k = \dfrac{T}{N} k$ を用いて書き直すと

$$
\begin{aligned}
f_k &= \sum_{n=0}^{N-1} W^{nk}\, \tilde{c}_n = \sum_{n=0}^{N-1} \tilde{c}_n \exp\left(\frac{2\pi i}{N} nk\right) = \sum_{n=0}^{N-1} \tilde{c}_n \exp\left(in \frac{2\pi}{T} \frac{T}{N} k\right) \\
&= \sum_{n=0}^{N-1} \tilde{c}_n \exp(in\omega t_k) \qquad (0 \le k \le N-1)
\end{aligned}
$$

となります．ここで角周波数 $n\omega$ の (複素) 単振動を表す関数

$$
e_n(t) = \exp(in\omega t)
$$

を導入すれば,

$$
f_k = \sum_{n=0}^{N-1} \tilde{c}_n e_n(t_k) \qquad (k = 0, 1, \ldots, N-1) \tag{3}
$$

となります.

ここで右辺の t_k を t にした関数

$$\sum_{n=0}^{N-1} \tilde{c}_n e_n(t) = \sum_{n=0}^{N-1} \tilde{c}_n \exp(in\omega t)$$

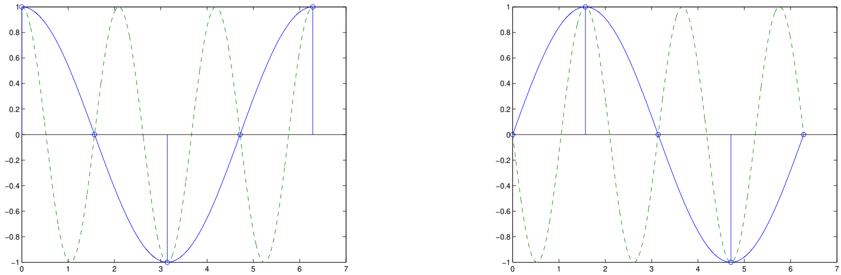
は有限複素フーリエ級数の形をしていますが，負の n に対応する項がなく，実数値関数の複素フーリエ展開の形はしていません．そこで

$$\begin{aligned}
\exp(i(n-N)\omega t_k) &= \exp\left(i(n-N)\frac{2\pi}{T}\frac{T}{N}k\right) = \exp\left(\frac{2\pi i}{N}(n-N)k\right) \\
&= \exp\left(\frac{2\pi i}{N}nk\right)\exp(-2\pi ik) = \exp\left(\frac{2\pi i}{N}nk\right) \\
&= \exp\left(in\frac{2\pi}{T}\frac{T}{N}k\right) = \exp(in\omega t_k)
\end{aligned}$$

より $e_{n-N}(t_k) = e_n(t_k)$ $(k = 0, 1, \ldots, N-1)$ が成立することに注意します．

2 つの関数 $e_{n-N}(t)$ と $e_n(t)$ は異なりますが，標本点 t_k における値は一致するため，標本化すると区別できなくなるというわけです．

たとえば $N = 4$ のとき，$e_1(t)$ （実線）と $e_{-3}(t)$ （点線）は標本点 $t_k = \dfrac{T}{4}k$ $(k = 0, 1, 2, 3)$ において同じ値を取るため，標本化すると区別できません．下の図は $e_1(t)$ と $e_{-3}(t)$ とその標本化を表したものです (左が実部，右が虚部).



実線 $e_1(t)$ と破線 $e_{-3}(t)$ は標本点で交わっています．このような現象をエイリアシング (aliasing) と呼びます．

定理 2.2（標本化定理）

$f(t)$ を周期 T の周期関数として $\omega = \dfrac{2\pi}{T}$ とおき，c_n を $f(t)$ の複素フーリエ係数とする．また N を自然数として区間 $[0, T]$ を N 等分し $t_k = \dfrac{k}{N}T$ として，$(\tilde{c}_0, \ldots, \tilde{c}_{N-1})$ を $(f(t_0), \ldots, f(t_{N-1}))$ の離散フーリエ変換とする．このときもし

$$n \geq \frac{N}{2} \quad \Rightarrow \quad c_n = 0 \tag{A}$$

ならば

$$0 \leq n < \frac{N}{2} \quad \Rightarrow \quad \tilde{c}_n = c_n \tag{B}$$

が成立する．

証明の前に仮定 (A) と結論 (B) の意味を考えましょう．M を $\dfrac{N}{2}$ より小さい最大の整数，すなわち N が偶数の時は $M = \dfrac{N}{2} - 1$, N が奇数のときは $M = \dfrac{N-1}{2}$ とすると，仮定 (A) は $f(t)$ の複素フーリエ級数が

$$f(t) = \sum_{n=-M}^{M} c_n e^{in\omega t} = c_0 + \sum_{n=1}^{M} (c_n e^{in\omega t} + \overline{c_n} e^{-in\omega t})$$

と表されることを意味します．これは $f(t)$ に含まれる単振動の角周波数 $n\omega$ が

$$n\omega \leq M\omega < \frac{N\omega}{2} = \frac{\pi N}{T} = \pi F_s$$

を満たすこと，すなわち $f(t)$ に含まれる単振動の周波数が $\dfrac{1}{2}Fs$ より小さいことを意味します（$F_s = \dfrac{N}{T}$ はサンプリング周波数）.

結論 (B) は

$$c_0 = \tilde{c}_0, \quad c_1 = \tilde{c}_1, \quad \ldots, \quad c_M = \tilde{c}_M$$

を意味するので，仮定 (A) と合わせて $f(t)$ のすべての複素フーリエ係数 c_n が $f(t)$ のサンプリングデータ $(f(t_0), f(t_1), \ldots, f(t_{N-1}))$ から離散フーリエ変換によって<u>正確に</u>求まります．従って $f(t)$ 自体が有限フーリエ級数

$$f(t) = \tilde{c}_0 + \sum_{n=1}^{M} (\tilde{c}_n e^{in\omega t} + \overline{\tilde{c}_n} e^{-in\omega t})$$

によって完全に復元できることになります．

以上をまとめると，標本化定理は，もとの関数 $f(t)$ がサンプリング周波数の $\dfrac{1}{2}$ 以上の周波数の単振動を含まなければ，このサンプリングによって情報は全く失われない，つまり $f(t)$ をサンプリングデータだけから完全に復元できることを意味しています．

標本化定理の証明：$e^{in\omega t_k} = \exp\left(in\frac{2\pi}{T}\frac{kT}{N}\right) = \exp\left(\frac{2\pi i}{N}nk\right)$ であるが，
$n < 0$ のときはさらに $\exp\left(\frac{2\pi i}{N}nk\right) = \exp\left(\frac{2\pi i}{N}(n+N)k\right)$ に注意する．
$f(t)$ の複素フーリエ展開の式に $t = t_k$ を代入して

$$
\begin{aligned}
f(t_k) = \sum_{n=-M}^{M} c_n e^{in\omega t_k} &= \sum_{n=-M}^{M} c_n \exp\left(\frac{2\pi i}{N}nk\right) \\
&= \sum_{n=0}^{M} c_n \exp\left(\frac{2\pi i}{N}nk\right) + \sum_{n=1}^{M} c_{-n} \exp\left(\frac{2\pi i}{N}(-n)k\right) \\
&= \sum_{n=0}^{M} c_n \exp\left(\frac{2\pi i}{N}nk\right) + \sum_{n=1}^{M} c_{-n} \exp\left(\frac{2\pi i}{N}(N-n)k\right) \\
&= \sum_{n=0}^{M} c_n \exp\left(\frac{2\pi i}{N}nk\right) + \sum_{n=N-M}^{N-1} c_{n-N} \exp\left(\frac{2\pi i}{N}nk\right)
\end{aligned}
$$

を得る．ここで最後の行ではその上の行の $N-n$ をあらためて n とした．

N が奇数のときは，$M = \dfrac{N-1}{2}$, $N - M = \dfrac{N+1}{2} = M + 1$ であり，

$$f_k = f(t_k) = \sum_{n=0}^{M} c_n \exp\left(\frac{2\pi i}{N} nk\right) + \sum_{n=M+1}^{N-1} c_{n-N} \exp\left(\frac{2\pi i}{N} nk\right)$$

となる．この式は $(c_0, c_1, \ldots, c_M, c_{-M}, c_{-(M-1)}, \ldots, c_{-1})$ の逆離散フーリエ変換が $(f_0, f_1, \ldots, f_{N-1})$ であることを意味している．従って，$(c_0, c_1, \ldots, c_M, c_{-M}, c_{-(M-1)}, \ldots, c_{-1})$ は $(f_0, f_1, \ldots, f_{N-1})$ の離散フーリエ変換である．一方，$(\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$ は $(f_0, f_1, \ldots, f_{N-1})$ の離散フーリエ変換であったから，

$$(c_0, c_1, \ldots, c_M, c_{-M}, c_{-(M-1)}, \ldots, c_{-1}) = (\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$$

が成立する．これから結論を得る．

一方，N が偶数のときは $M = \dfrac{N}{2} - 1$, $N - M = 2(M+1) - M = M+2$ より，

$$f_k = f(t_k) = \sum_{n=0}^{M} c_n \exp\left(\frac{2\pi i}{N} nk\right) + \sum_{n=M+2}^{N-1} c_{n-N} \exp\left(\frac{2\pi i}{N} nk\right)$$

となるから，上と同様の議論により，

$$(c_0, c_1, \ldots, c_M, 0, c_{-M}, \ldots, c_{-1}) = (\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$$

が成立することがわかる,（証明終わり）

例：$T>0$, $\omega=\dfrac{2\pi}{T}$ とおく．区間 $[0,T]$ を 4 等分し $t_k=\dfrac{kT}{4}$ $(k=0,1,2,3)$ として周期 T の関数 $f(t)=\sin^3\omega t$ をサンプリングして離散フーリエ変換を求めると ($\omega t_k=\dfrac{k\pi}{2}$ に注意）$\sin\omega t$ のときと同様に

$$(f_0,f_1,f_2,f_3)=(0,1,0,-1),\qquad (\tilde{c}_0,\tilde{c}_1,\tilde{c}_2,\tilde{c}_3)=\Big(0,-\frac{i}{2},0,\frac{i}{2}\Big)$$

一方 $f(t)$ の複素フーリエ級数展開は（スライド 7 の例 1.8）

$$\begin{aligned} f(t)=\sin^3\omega t&=\Big(-\frac{i}{2}\Big)^3(e^{i\omega t}-e^{-i\omega t})^3\\ &=-\frac{3i}{8}e^{i\omega t}+\frac{i}{8}e^{3i\omega t}+\frac{3i}{8}e^{-i\omega t}-\frac{i}{8}e^{-3i\omega t}\end{aligned}$$

なので，$f(t)$ の複素フーリエ係数 c_n は

$$c_0=0,\quad c_1=-\frac{3i}{8},\quad c_2=0,\quad c_3=\frac{i}{8},\quad c_n=0\quad(n\neq\pm1,\pm3)$$

となり，$c_1\neq\tilde{c}_1$, $c_3\neq\tilde{c}_3$ となります．よって 1 周期を $N=4$ 等分したサンプリングではもとの関数 $f(t)$ を復元できません．では N をいくつ以上にとれば，サンプリングから $f(t)$ を復元できるでしょうか？（例終わり）

以上により，アナログ信号 $f(t)$ を標本化する前に，$f(t)$ に含まれる周波数が $\frac{F_s}{2}$ 以上の成分はあらかじめ除去しておけば（これはローパスフィルター呼ばれる電気回路で実現できます），サンプリングによって元のアナログ信号のスペクトルが正確に決定でき，従って元の信号を完全に復元できることがわかりました.

実際の音声解析では，まず音声信号 $f(t)$ を $\frac{F_s}{2}$ 以上の周波数成分をカットするような回路 (Low Path Filter) を通してから標本化 (および値を有限個の数で近似する量子化と呼ばれる操作) を行い，次にそのデータ (ベクトル) を離散フーリエ変換して，$(\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$ を求めます．このうち $0 \leq n < \frac{N}{2}$ の範囲の n に対する $\tilde{c}_n$ が，もとの音声信号 $f(t)$ の複素フーリエ係数 c_n と一致します。

よって，

$$|\tilde{c}_0|,\ |\tilde{c}_1|,\ \ldots,|\tilde{c}_{\frac{N}{2}-1}| \quad (\text{または } |\tilde{c}_{\frac{N-1}{2}}|)$$

が $f(t)$ のスペクトルを表します．($|\tilde{c}_n|$ は $f(t)$ に含まれる角周波数 $n\omega$ の単振動の振幅を表します．)

たとえば，$f(t)$ が周波数 441Hz の音声信号を表しているとしましょう．サンプリング周波数を $F_s = 44100\text{Hz}$ とすると，1 周期 $T = \dfrac{1}{441}$ を 100 等分した t_k に対して $f(t_k)$ の値を記録することになるので，$N = 100$ となります．従って，もし $f(t)$ が 22050Hz 以上の周波数成分を含んでいなければ，$\omega = 2\pi \cdot 441$ として

$$f(t) = \tilde{c}_0 + 2\mathrm{Re}\left\{\sum_{n=1}^{49} \tilde{c}_n e^{in\omega t}\right\} = \tilde{c}_0 + \sum_{n=1}^{49} \tilde{c}_n e^{in\omega t} + \sum_{n=1}^{49} \overline{\tilde{c}_n} e^{-in\omega t}$$

が $f(t)$ の有限フーリエ級数による表示になります．

2.3 MATLAB による音声解析

1. 離散フーリエ変換の計算

15 ページの例 $(\sin^3 \omega t)$ の計算を MATLAB で行ってみましょう．簡単のため $\omega = 1$ とすると周期は $T = 2\pi$ となるので，区間 $[0, 2\pi]$ を 8 等分してサンプリングしてみましょう．

まず，添字 $k = 0, 1, 2, \ldots, 7$ をベクトル $(0, 1, \ldots, 7)$ として

```
k = 0:1:7;
```

で定義します．すると $x_k = \dfrac{2\pi}{8} k = \dfrac{\pi}{4} k \ (0 \leq k \leq 7)$ もベクトルとして

```
x = (pi/4)*k;
```

で定義できます．

$f(x) = \sin^3 x$ のサンプリングは

```
f = sin(x).^3;
```

で求まります．セミコロンなしで `f` と入力すれば $(f_0, f_1, \ldots, f_7)$ が表示されます.

この離散フーリエ変換は

```
c = (1/8)*fft(f)
```

で計算できます．fft とは離散フーリエ変換を高速に計算する (Fast Fourier Transform) 関数です．ただし授業の定義と違って N で割る前の値を返します．そこで 1/8 を掛けました.

2. 関数の音を聴く

$f(t)$ を時刻 t (秒単位) の関数とするとき，$f(t)$ の音を聴いてみましょう．CD などのデジタル録音では，1 秒間を 44100 等分した t の値に対する $f(t)$ の値を成分とするベクトルを作ります．44100 という数字 (サンプリング周波数という) は頻繁に出てくるので，それを最初に Fs という名前で定義しておきます．たとえば 1 秒間音を鳴らすためには，区間 $[0, 1]$ を Fs 等分してできるベクトルを t とします．

```
Fs = 44100;
t = linspace(0,1,Fs+1);
```

440Hz の単振動 $y = \sin(2\pi \cdot 440t)$ は

```
y = sin(2*pi*440*t);
```

で定義できます．この音を聴くには次のようにします.

```
soundsc(y,Fs);
```

もしここでスペルミス以外のエラーが表示された場合は,

```
restoredefaultpath; matlabrc
savepath
```

を 1 行ごとにリターンキーで実行してください。

ドミソの和音を聴くには，たとえば関数名を ceg とすると

```
ceg = sin(2*pi*264*t)+sin(2*pi*330*t)+sin(2*pi*396*t);
soundsc(ceg,Fs);
```

3. 音の録音と波形・スペクトルの表示

まず録音の準備をします．システム環境設定 → サウンド → 入力として，マイクが働いていることを確認し，感度を調整してください．次に，

```
Fs = 44100;
r = audiorecorder(Fs,16,1);
```

とします．最初の行でサンプリング周波数を指定します．2 行目の命令は，これから録音するデータに `r` という名前を付けて保存することを意味します．(Fs,16 はサンプリング周波数とデータのビット数の指定，1 はチャンネル数．) 以上の 2 つの命令は最初に 1 度だけ入力すれば十分です．

実際に録音を行うには，たとえば

```
record(r,2);
```

とします．2 は録音時間（秒単位）を指定しています．場合に応じて 1 や 3 など適当に変えてください．これを入力してリターンキーを押すと，その瞬間からこの場合は 2 秒間の音が録音されて，`r` という変数に格納されます．録音できたかどうか確かめるには

```
play(r);
```

として再生します．`r` のデータは関数などの表示には使えないので，次のようにして通常のデータに変換する必要があります．`double` は倍精度小数を表します．

```
v1 = getaudiodata(r,'double');
```

これで今録音したデータが `v1` という名前のベクトルに格納されました．(`v1` の代わりに好きな名前を付けてください．何回か録音する場合は，上の命令で `v1` のかわりに v2, v3 などとすればいくつもの音声データを扱えます．) `v1` の大きさ（成分数）は `Fs` と録音時間（秒）との積ですが，

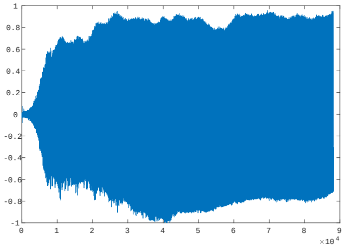
`size(v1)` または `length(v1)`

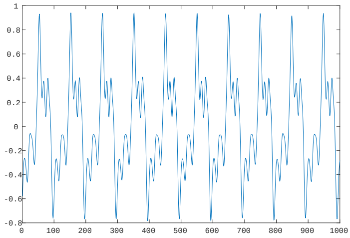
として確認することもできます．なお MATLAB ではベクトルや行列の添字は（0 ではなく）1 から始まるので，たとえば，`v1` の最初の成分 (添字が 0) を見るには `v1(1)` とします．

さて，v1 のグラフを表示させて見ましょう．

```
plot(v1);
```

v1 は $2 \times F_s = 88200$ 個の要素からなるベクトルですから，実際の波形を見るにはこのベクトルの 1 部分を取り出す必要があります．このグラフの横軸は要素の番号 (1 から 88200 まで) を表しています (下左図).





波形がなるべく安定しているところを見つけて，少しずつ拡大していきます．たとえば，まず 70001 番目から 71000 番目までを表示させるには，

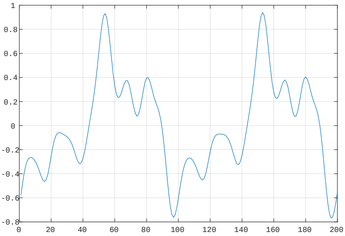
```
plot(v1(70001:71000));
```

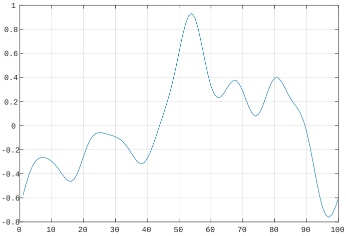
とします (上右図)．この図の横軸は 70001 番目を新たに 1 番目とした表示であることに注意してください．さらに

```
plot(v1(70001:70200)); grid on;
```

として，70001 番目から 70200 番目までを取り出してみましょう．grid on で網目が表示されます (下左図)．最後にちょうど 1 周期分を取り出します (下右図)．

```
plot(v1(70001:70100)); grid on;
```





この 1 周期分のデータに名前 (ここでは `f` とします）を付けておきましょう.

```
f = v1(70001:70100);
```

この場合 `f` は大きさ (次元) N が 100 のベクトルになります．実際の音声では N は周波数に応じて異なる数字になります．このベクトル $f = (f_0, f_1, \ldots, f_{N-1})$ が 1 周期分の音のサンプリングデータです．ただし MATLAB では添字が 1 から始まるので

$$f_0 = \mathtt{f(1)}, \quad f_1 = \mathtt{f(2)}, \quad \ldots, \quad f_{N-1} = \mathtt{f(N)}$$

となることに注意してください．周波数は F_s/N （この例では 441Hz) となります.

以下で利用するため，この例の場合では `N = 100;` としておきます．スペクトルのグラフで添字が 1 ではなく 0 から始まるようにするために，

```
k = 0:1:(N-1);
```

として 0 から $N-1$ までを成分とするベクトル `k` を定義します．次に

```
c = fft(f);
```

として `f` の離散フーリエ変換 `c` を計算します．授業での $(\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1})$ に当たります．ただし MALTAB の `fft` は N で割りません．しかしこれはスペクトルの形には影響しないので（値には影響しますが），気にしないことにしましょう．

次に

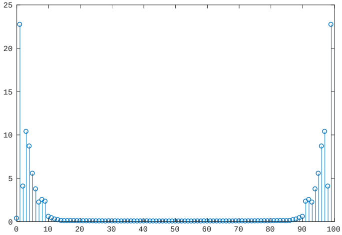
```
stem(k,abs(c));
```

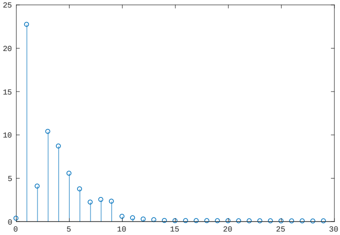
として c の絶対値の数列（スペクトル）$(|\tilde{c}_0|, |\tilde{c}_1|, \ldots, |\tilde{c}_{N-1}|)$ を棒グラフで表示します (下左図).

標本化定理により，離散フーリエ変換の左半分，すなわち $[(N-1)/2]$ 番目までが，実際の音声の複素フーリエ係数を表していることを思い出しましょう．たとえば 1 番目 $(|\tilde{c}_0|)$ から 31 番目 $(|\tilde{c}_{30}|)$ までを表示させるには

```
stem(k(1:31),abs(c(1:31)));
```

とします (下右図).


25
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100


25
20
15
10
5
0
0
5
10
15
20
25
30

4. データの保存と呼び出し

データの保存をするには，保存するファイル名と保存する変数名を指定します．たとえば

```
save('voice.mat','v1','f','c');
```

とすると，データ `v1`, `f`, `c` がまとまって，'voice.mat' という名前のファイルに保存されます．ここで `voice` は好きな文字列に置き換えて良いですが，拡張子 `.mat` の部分はこの通りにしてください．MATLAB で作成したファイルは，書類フォルダの中の MATLAB というフォルダの中に保存されます．

後でこのファイルをロードするには，MATLAB のフォルダウィンドウ（左の窓）から目的のファイルを選んでダブルクリックするか，または上にある「開く」タブからファイルを選びます．ファイルをロードすると右のワークスペースに呼び出された変数の名前 `v`, `f`, `c` が表示され，以後これらの変数を使用することができます．

5. 簡単なプログラミング

離散フーリエ変換から標本化定理を用いて元の音声信号を有限フーリエ級数で「復元」するプログラムを作成しましょう.

(1) メニューバーの「新規」タブから「関数」選んでクリックすると、別の窓 (Editor) が開くので，そこに次のプログラムを入力します.

```
function y = fourier(c,omega,K)
Fs = 44100;
t = linspace(0,1,Fs+1);
y = zeros(1,Fs+1);
for k=1:1:K
y = y + c(k+1)*exp(k*i*omega*t);
end
y = c(1) + 2*real(y);
end
```

ここで i は虚数単位 $\sqrt{-1}$ を表します．t は区間 $[0,1]$ を F_s 等分してできる分点（右端も含めて F_s+1 個）からなるベクトルを表します．

(2) プログラムの入力が完了したら、メニューバーの「保存」から「保存」を選んでクリックします。名前は自動的に '関数名.m'（この場合は fourier.m）となるはずですが，そうでなければ手動で入力または修正します．(拡張子 .m を忘れないように).

(3) 離散フーリエ変換 c を用いてたとえば 49 項目までの有限フーリエ級数を作り，y という変数に保存するには，もし音の周波数が 441Hz ならば，Command Window で

```
y = fourier(c,2*pi*441,49);
```

と入力します．このフーリエ級数の音を聴くには，

```
soundsc(y,Fs);
```

とします.

関数 fourier(c,omega,K) は，c が $(c_0, c_1, \ldots, c_{N-1})$ というベクトルを表すとき，角周波数が $\omega =$ omega の複素フーリエ級数の第 K 項目までの和（ただし $K < N/2$)

$$f_K(t) = c_0 + \sum_{n=1}^{K} c_n e^{in\omega t} + \sum_{n=1}^{K} \overline{c_n} e^{-in\omega t} = c_0 + 2\mathrm{Re}\left(\sum_{n=1}^{K} c_n e^{in\omega t}\right)$$

を計算します．c としては離散フーリエ変換 fft の計算結果を使います．標本化定理によって，$0 \leq n < N/2$ のとき離散フーリエ変換 $\tilde{c}_n$ は複素フーリエ係数 c_n と一致するからです．特に $K = [(N-1)/2]$ とすれば，$f_K(t)$ は，1 周期分のデータ `f` を周期的に拡張して 1 秒間に伸ばした音を表しています．`omega` を変えれば，もともと異なる周波数の音にすることもできます．