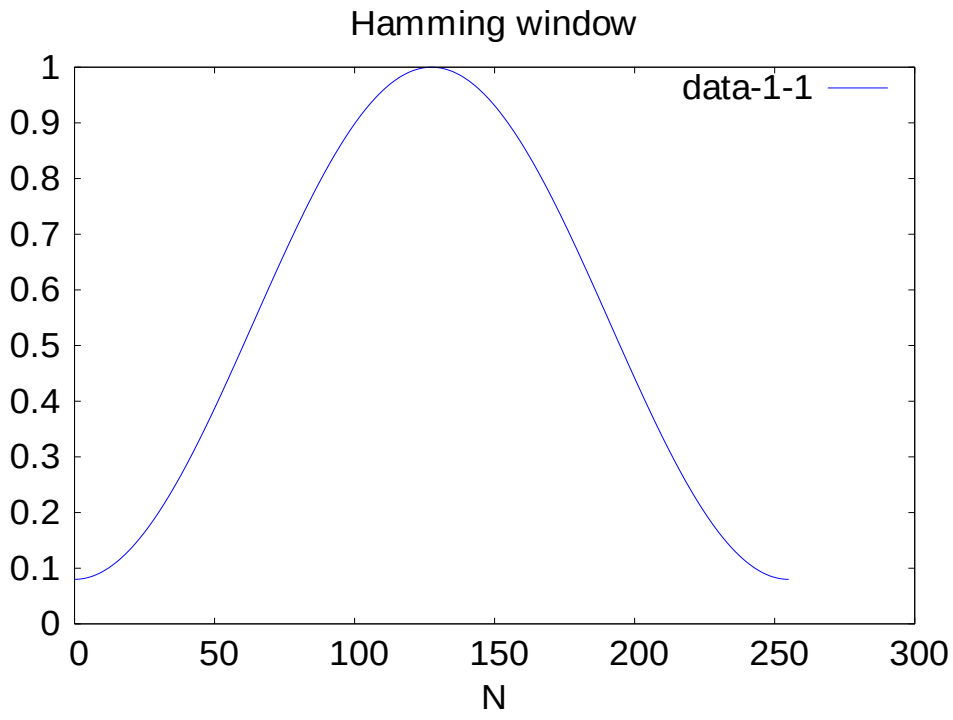
７．ハミング窓とフラットトップ窓の等価ノイズ帯域幅（ENBW）

（１） Hamming 窓

Hamming 窓は次式で表されます。MaTX にも関数が用意されています。

win = 0.54- 0.46*cos(2*PI*[k/(N-1)); ただし k=0,1,---,N-1

```
N=256;
K=[0:N-1];
w=0.54-0.46*cos(2*PI*K/(N-1));
mgplot_reset(1);
mgplot_title(1,"Hamming window");
mgplot_xlabel(1,"N");
mgplot_ylabel(1,"Amplitude");
mgplot(1,K,w);
sum_sq_w=sum(w.*w)
sq_sum_w=(sum(w))^2
AmplitudeCorrectionFactor=sum(w)/N
Power_cf=sum_sq_w/N
ENBW_cf=sum_sq_w/sq_sum_w*N
```



sum_sq_w = 101.343

sq_sum_w = 18983.3

AmplitudeCorrectionFactor = 0.538203　　NI の資料によると 0.54 です。

Power_cf = 0.395873

ENBW_cf = 1.36667　（N=2048 なら 1.3633）　NI の資料によると 1.36 です。

（2）フラットトップ窓

フラットトップ窓は、小野測器の Web など日本の資料と欧米の Matlab や National instruments 社の資料とでは　定義の数式が異なります。

日本の数式：

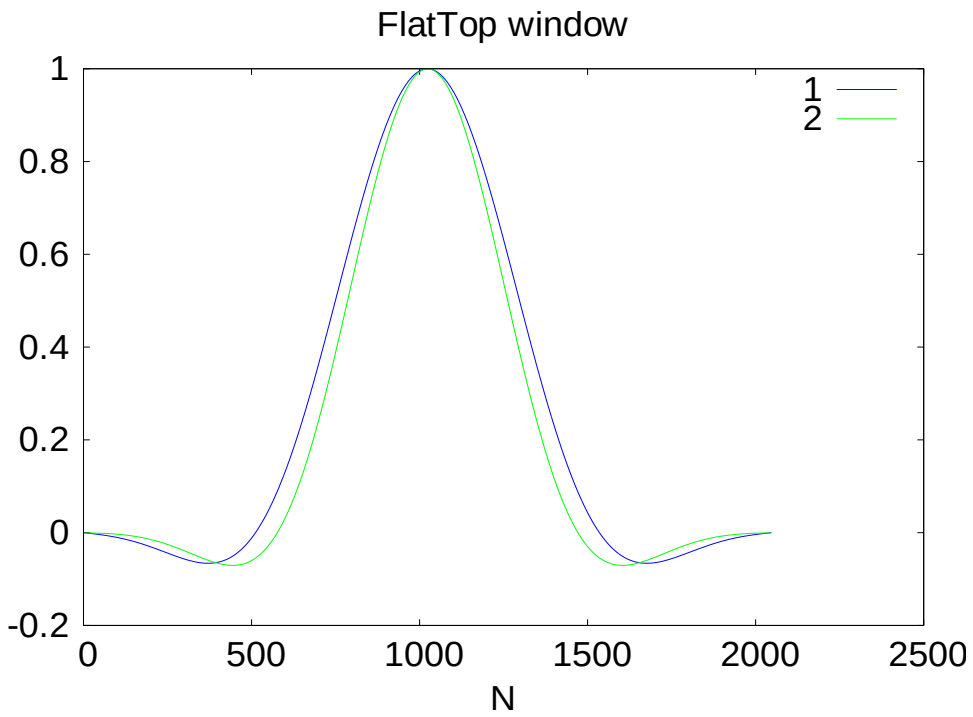
win=(0.54+0.46*cos(2*PI*(k-(N-1.0)/2.0)/(N-1.0))*sin(4*PI*(k-N/2.0)/N)/(4*PI*(k-N/2.0)/N)

米国の数式：win=(1-1.933*cos(2*PI*k/N)+1.286*cos(4*PI*k/N)-0.388*cos(6*PI*k/N)
+0.0322*cos(8*PI*k/N))/(1+1.933+1.286+0.388+0.0322);

ただし、k=0,1,---,N-1

両者を比較してみます。尚、米国の数式の分母(1+1.933+1.286+0.388+0.0322)は最大値を 1 にそろえて比較するために追加しました。

```
N=2048;
k=[0:N-1];
w=4*PI*(k-(N-1.0)/2.0)/(N-1.0);
w1=(0.54+0.46*cos(0.5*w)*sin(w)/w;    //日本の数式
w2=(1-1.933*cos(2*PI*k/N)+1.286*cos(4*PI*k/N)-0.388*cos(6*PI*k/N)+
0.0322*cos(8*PI*k/N))/(1+1.933+1.286+0.388+0.0322);    //欧米の数式
mgplot_reset(1);
mgplot_title(1,"FlatTop window");
mgplot_xlabel(1,"N");
mgplot_ylabel(1,"Amplitude");
mgplot(1,k,[[w1][w2]],{"1","2"});
sum_sq_w1=sum(w1.*w1)
sq_sum_w1=(sum(w1))^2
AmplitudeCorrectionFactor1=sum(w1)/N
Power_cf1=sum_sq_w1/N
ENBW_cf1=sum_sq_w1/sq_sum_w1*N
sum_sq_w2=sum(w2.*w2)
sq_sum_w2=(sum(w2))^2
AmplitudeCorrectionFactor2=sum(w2)/N
Power_cf2=sum_sq_w2/N
ENBW_cf2=sum_sq_w2/sq_sum_w2*N
```



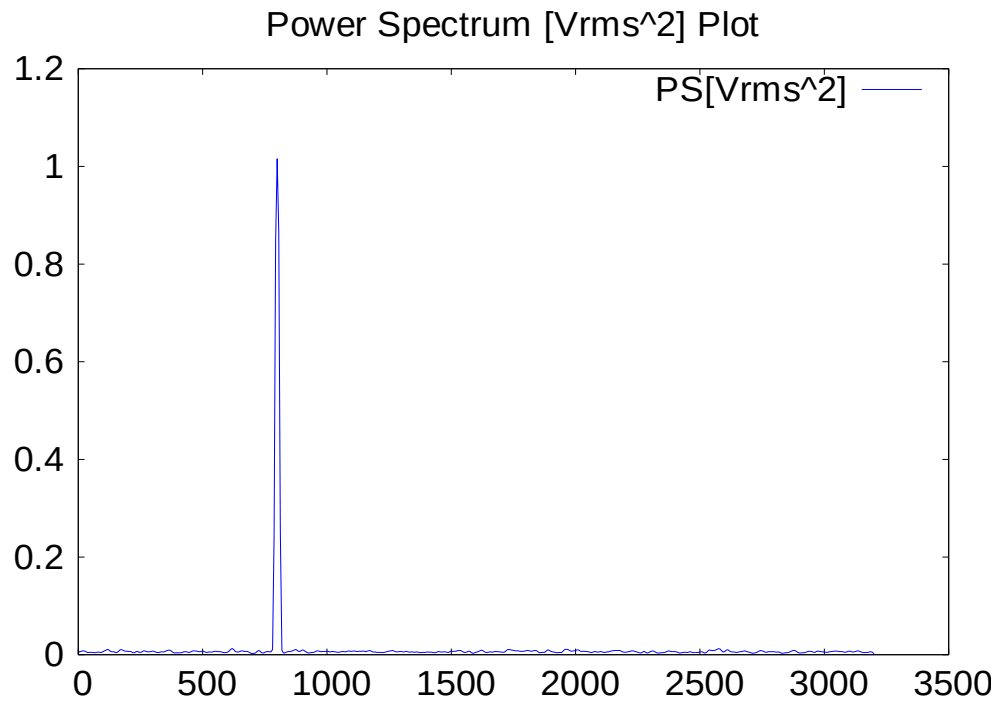
１が日本の数式、２が欧米の数式です。両者はよく似ていますが、微妙に差があります。

sum_sq_w1 = 412.214
sq_sum_w1 = 263942
AmplitudeCorrectionFactor1 = **0.250856**
Power_cf1 = 0.201276
ENBW_cf1 = **3.19849**
sum_sq_w2 = 358.833
sq_sum_w2 = 194883
AmplitudeCorrectionFactor2 = **0.215554**
Power_cf2 = 0.175212
ENBW_cf2 = **3.77093**

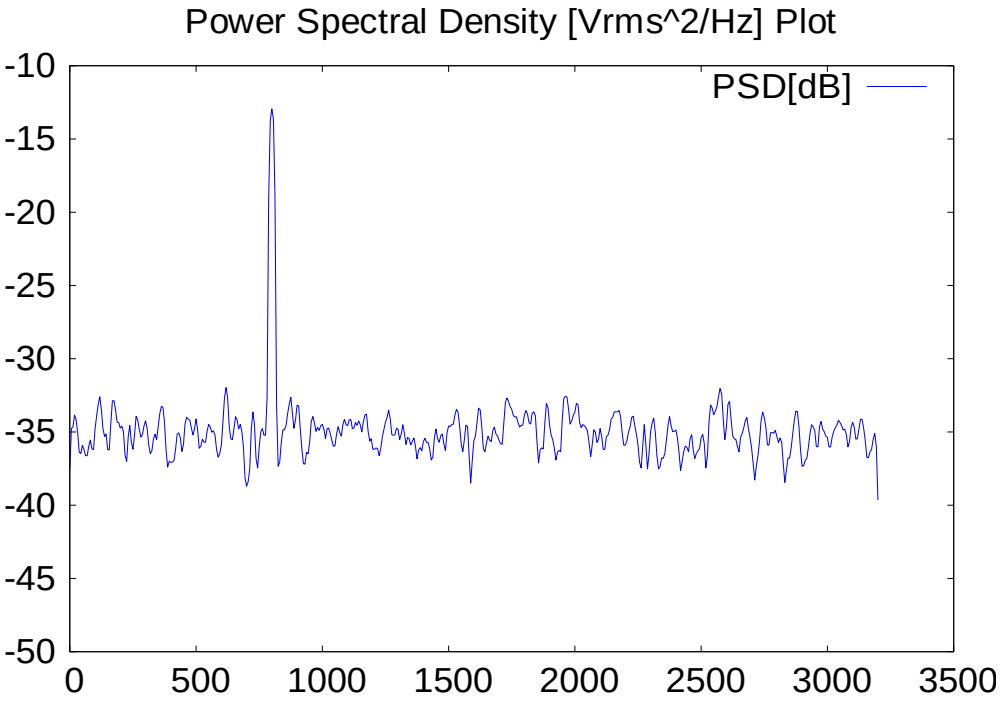
NI 社の資料によると、米国の数式の振幅補正係数は 0.22、等価ノイズ帯域幅補正係数は 3.77 とされており、計算と一致します。
日本の数式の場合は、小野測器の資料では等価ノイズ帯域幅補正係数が 3.67 と記載されており、計算値 3.20 と一致しません。

そこで、日本の数式について振幅補正係数 0.25、等価ノイズ帯域補正係数 3.20 が妥当であることを検証するため、「1Vrms^2 のホワイトノイズ＋1Vrms の 800Hz 正弦波」を対象にフラットトップ窓を使ってパワースペクトラムを求めてみます。
MaTX のプログラムは PSD734.mm（末尾に掲載）です。

振幅補正係数を使ってパワースペクトラムを求めると、800Hz の振幅が確かに 1Vrms^2 となっており、振幅補正係数は妥当と確認できます。（800Hz 成分計算値は 1.01569Vrms^2 でした。）



次に等価ノイズ帯域幅補正係数を使って Vrms^2/Hz 単位のパワースペクトラムを求めると、ノイズレベルが 1Vrms^2/3200Hz=-35dB になっており、この補正係数も妥当です。



パワーの総和の計算値は、理論値 2Vrms^2 に対し、下記のように一致しています。

Pgaus = 2.01863　　フラットトップ窓を掛ける前のパワー
Ppsd = 2.01409　　帯域幅補正を使ったパワースペクトラムから求めたパワー

参考文献

The Fundamentals of FFT-Based Signal Analysis and Measurement in LabVIEW and LabWindows/CVI

http://zone.ni.com/devzone/cda/tut/p/id/4278

KTH（スウェーデン王立工科大学）MWL の Signal Analysis 第 9 章
http://www.flyg.kth.se/education/msce/soundvib/courses/SD2130/index.html

Matlab の Sinal Processing Toolbox
http://dl.cybernet.co.jp/matlab/support/manual/r14/toolbox/signal/?/matlab/support/manual/r14/toolbox/signal/flattopwin.shtml

小野測器 「FFT アナライザについて」の 7 章
http://www.onosokki.co.jp/HP-WK/c_support/newreport/analyzer/FFT4/fft_13.htm

疑問点
Hanning 以外の窓についても、同様に補正量を計算できます。ここで、フラットトップの場合はちょっと疑問です。と言うのは窓のカーブが端の方で負になる部分を含むためです。負の部分を引き算して良いのか？ノイズはランダムなので、自乗してパワーで取り扱うのに、振幅を例えば、絶対値で扱わなくてよいのか？

振幅は信号成分の大きさを見るためで、信号のある周波数成分は一定の振幅と位相を有しており、窓関数に負の部分があると、逆位相になって振幅が打ち消されるので、絶対値でなく正負の符号をそのまま扱う方が正しいと言えます。
ノイズのパワーは自乗で計算しているので、窓関数の正負は関係なくなります。
測定時間幅の周期を有する信号とランダムノイズの和からなる観測対象をパワースペクトルで取り扱う場合を考えてみます。信号は測定時間幅の逆数の周波数の高調波からなる線スペクトルの集まりです。各周波数成分はある一定の振幅と位相をもつので、窓関数を掛けるとき、窓関数に負の部分があればその部分は逆位相になって、信号の大きさは打ち消されて小さくなります。従って、素直に正負の符号をそのまま使ってよいと考えられます。すなわち振幅補正（Amplitude correction factor）の考え方は窓関数に負の領域があっても扱いは同じでよいと考えられます。
ノイズについてはある各周波数成分の位相は一定ではなく測定の度に違いランダムですから、パワー（自乗和）としてしか扱えません。したがってパワー補正（Power correction factor）の扱いも正しいと考えられます。
ENBW は振幅補正した信号に対する補正であり、窓関数を加える前の原信号に対する補正量は Power cf に還元されます。

PSD734.mm

```
//PSD_test734
//2008 11.24, 2009.03.21 Y.Sugimoto
Fs=6400.0;      // sampling frequency
N=1024;          //1回の FFT のサンプル数
```

```
M=16*N;            //FFT を 16 回繰り返して平均を求めるため
Freq_step=Fs/N;
t=[0:M]/Fs;     //時刻列（配列）
tm=t(1:M);
randn("seed",1);
xn=randn(t)+sqrt(2)*sin(1600*PI*t);
//xn=sqrt(2)*sin(1600*PI*t);
//M 個のガウシアンノイズ（平均パワー1Vrms^2）と 1Vrms^2 の 800Hz 正弦波
//平均パワーを確認  2Vrms^2 になるはず
Pgaus=sum(xn(1:M)*xn(1:M))/M

/*乱数で作ったノイズをフィルタを通さずに使っているので、
矩形窓でもよいが、ENBW_cf などを考慮に入れた計算を試すため
Flat Top 窓を使う。*/
k=[1:N];

w=4*PI*(k-(N-1.0)/2.0)/(N-1.0);
w=(0.54+0.46*cos(0.5*w))*sin(w)/(w);   //日本のフラットトップ窓

//w=1-1.933*cos(2*PI*k/N)+1.286*cos(4*PI*k/N)-
//        0.388*cos(6*PI*k/N)+0.0322*cos(8*PI*k/N);
//w=w/(1+1.933+1.286+0.388+0.0322);//欧米のフラットトップ窓

sum_sq_w=sum(w*w);
sq_sum_w=(sum(w))^2;
Amp_cf=sum(w)/N
Power_cf=sum_sq_w/N;
ENBW_cf=sum_sq_w/sq_sum_w*N

//最初の4回の測定におけるパワーのバラツキをチェック
P1=sum(abs(fft(w*xn(1:N)))^2)/(N^2)/Amp_cf^2/ENBW_cf
P2=sum(abs(fft(w*xn(N+1:2*N)))^2)/(N^2)/Amp_cf^2/ENBW_cf
P3=sum(abs(fft(w*xn(2*N+1:3*N)))^2)/(N^2)/Amp_cf^2/ENBW_cf
P4=sum(abs(fft(w*xn(3*N+1:4*N)))^2)/(N^2)/Amp_cf^2/ENBW_cf

AvPxx=(abs(fft(w*xn(1:N)))^2+
        abs(fft(w*xn(N+1:2*N)))^2+
        abs(fft(w*xn(2*N+1:3*N)))^2+
        abs(fft(w*xn(3*N+1:4*N)))^2+
        abs(fft(w*xn(4*N+1:5*N)))^2+
        abs(fft(w*xn(5*N+1:6*N)))^2+
        abs(fft(w*xn(6*N+1:7*N)))^2+
        abs(fft(w*xn(7*N+1:8*N)))^2+
        abs(fft(w*xn(8*N+1:9*N)))^2+
        abs(fft(w*xn(9*N+1:10*N)))^2+
```

```
        abs(fft(w*xn(10*N+1:11*N)))^2+
        abs(fft(w*xn(11*N+1:12*N)))^2+
        abs(fft(w*xn(12*N+1:13*N)))^2+
        abs(fft(w*xn(13*N+1:14*N)))^2+
        abs(fft(w*xn(14*N+1:15*N)))^2+
        abs(fft(w*xn(15*N+1:16*N)))^2)/(N^2*16)/Amp_cf^2;

f=Freq_step*[0:N-1];
AvPps=sum(AvPxx)/ENBW_cf
//フーリエ変換後の周波数成分の自乗和からも同じ値が得られる。

PS=2*AvPxx(1:N/2+1);
PS(1)=AvPxx(1);
PS(N/2+1)=AvPxx(N/2+1);
/*PS は FFT アナライザの表示に対応したパワースペクトル密度
ただし単位は Vrms^2
ノイズパワー1Vrms^2 が N/2 個の周波数サンプルに分散、ただし
等価帯域幅補正係数分(=3.2)大きく表示される
3.2/(N/2)=3.2/512→－22.0dB
PS を等価帯域幅=周波数刻み*ENBW_cf で割れば、Vrms^2/Hz 単位になる。
ノイズの場合は線スペクトルではないので Vrms^2/Hz で扱う方が合理的
*/
PSD=PS/(Freq_step*ENBW_cf);
Pps=sum(PS)/ENBW_cf
Ppsd=sum(PSD)*Freq_step
/*ホワイトノイズなら sum(PSD)*Freq_step=PSD*N/2*Δf=PSD*ナイキスト周波
数
ホワイトノイスが 1Vrms^2 なら、PSD は 3.2kHz で割って、
312.5e-6Vrms^2/Hz=-35dB
*/
mgplot_reset(1);
mgplot_yrange(1, 0.0, 1.2);
mgplot_title(1,"Power Spectrum [Vrms^2] Plot");
mgplot(1,f(1:N/2+1),PS,{"PS[Vrms^2]"});
//振幅は単一スペクトルなら実効値の自乗になる
//800Hz 成分とその前後の成分の大きさをチェック
PS(127)
PS(128)
PS(129)
PS(130)
PS(131)
pause;
mgplot_reset(2);
mgplot_yrange(2, -30.0, 10.0);
mgplot_title(2,"Power Spectrum [Vrms^2] Plot");
```

```
mgplot(2,f(1:N/2+1),10*log10(PS),{"PS[dB]"});
//ノイズの振幅は-22dB
pause;
mgplot_reset(3);
mgplot_yrange(3, -50.0, -10.0);
mgplot_title(3,"Power Spectral Density [Vrms^2/Hz] Plot");
mgplot(3,f(1:N/2+1),10*log10(PSD),{"PSD[dB]"})
//ノイズの振幅は-35dB
```