ME525 Applied Acoustics Lecture 15, Winter 2024
Spectral Analysis Part I

Peter H. Dahl, University of Washington

We switch gears aover the next couple of lectures towards getting a practical, working knowledge of FFT-based analyis of acoustic data. Figure 1 shows a recording of 10 sec of underwater pressure data recorded at a depth of about 500 m in the Western Pacific. The sample rate (or sample frequency) is f_s 1600 Hz, meaning each sample is separated by $dt = 1/f_s$ sec.

The Nyquist sampling theorem tells us that if you discretely sample a time series of anything (e.g., earthquake seismic recording, music, a heart EKG signal, the stock market) with time separation equal to $dt = 1/f_s$, then the maximum observable frequency in any subsequent analysis, e.g.., spectrum analysis, $f_s/2$. As a consequence, if your time series possesses energy, or content, at frequencies greater than $f_s/2$, the resulting discrete time series associated with your digital recording based $dt = 1/f_s$, is aliased. At this point there is not much you can do with the corrupted data and thus we take great pains to avoid aliasing. One way is to increase f_s (decrease dt) but this can result in a large data storage. Another way is to pre-filter the data with an anti-aliasing filter to remove all frequency content greater than $f_s/2$ before undertaking the digital sampling. From the Nyquist theorem, you should thus understand that we can studyt frequency content in the Fig. 1 data up to $\leq 0.5 f_s$ or 800 Hz.

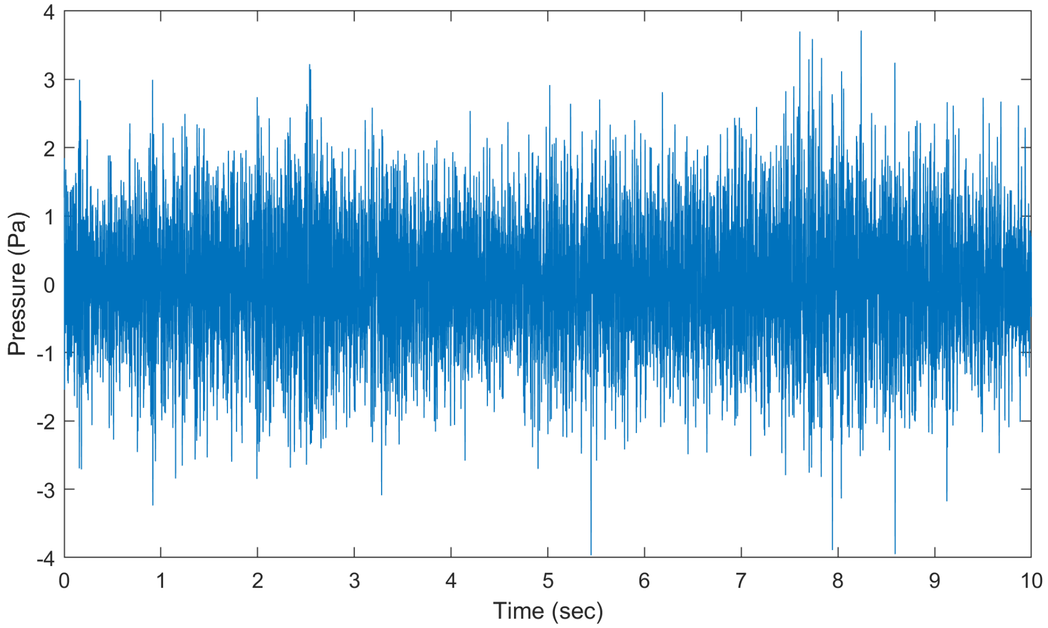

Figure 1: Underwater sound recording made at depth in the Western Pacific at a depth of 500 m.

In matlab the the data in Fig. 1 is named `data_Pa`, which is of length N = 16000 points, and a full-length FFT of this data is `X=fft(data_Pa)`, with result shown in Fig. 2. Notice the symmetry about the half-way point. If you needed to work with negative frequencies (see supplementary notes), then plot `fftshift(fft(data_Pa))`, but this is not our current interest. Instead we retain only the first half of the FFT vector `X = X(1:N/2)` (for simplicity best to have N even, but not essential). Make a corresponding 800-point frequency vector in matlab as follows: `f = (0:1:(0.5N-1))`, and then insure the highest frequency equals $f_s/2$, or 800 Hz, with `f = (f/f(end))*fs/2`.

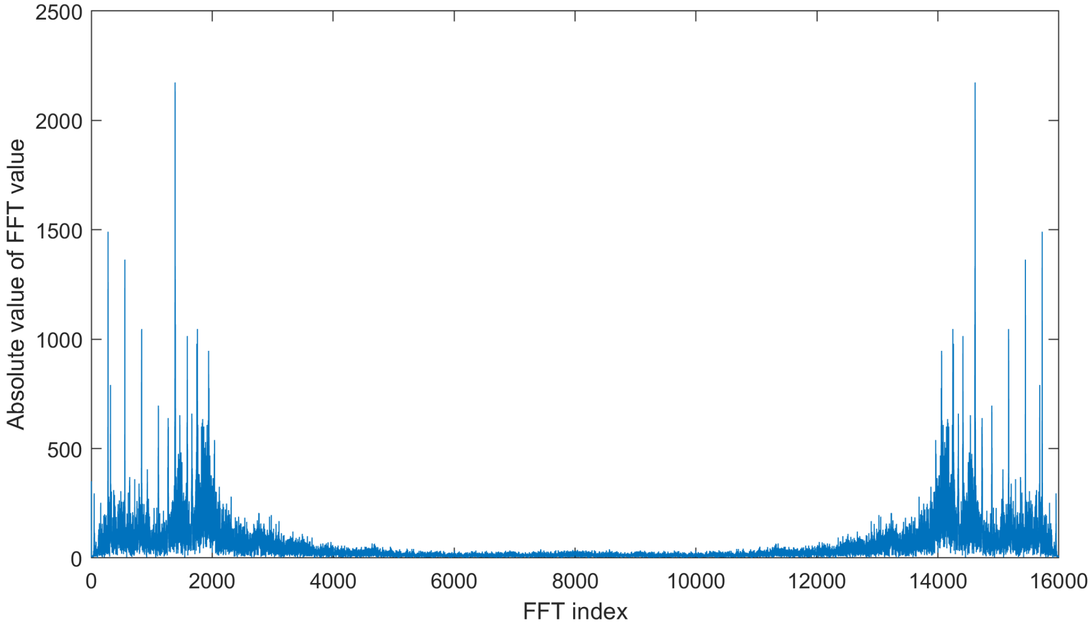

Figure 2: Plot of the absolute value of the 16000 complex FFT coefficients X corresponding to the data in shown in Fig. 1.

Notation can be variable, but here define the discrete representation X as a function of f, as computed above, where X and f are each length 800; and $X(f)$ is complex-valued. Now define

$$S(f) = \frac{2dt}{N}|X(f)|^2 \tag{1}$$

Equation (1) gives the proper normalization for integrating $S(f)$ over the frequency range 0 to f_{max} = 800 Hz, to yield the variance, σ^2 or mean-square value, of the original data in the time domain. The proper dimension of $S(f)$ (Fig. 3) is Pa^2/Hz, and thus $S(f)$ might also be referred to as a variance spectrum. Note that one can force the normalization as follows

$$S(f) = \sigma^2|X(f)|^2 / \int_0^{f_{max}} |X(f)|^2 df \tag{2}$$

Although both Eqs. (1) and (2) yield the same result, it is good to keep Eq.(1) for a better understanding on the role of dt and N.

Two versions of $S(f)$ are displayed in Fig. 3, one labeled $\nu = 2$ is that which we have been discussing, based on a FFT over the entire length of the data, or $K = 1$. The one labeled $\nu = 20$, is computed by segmenting the data in Fig. 1 into K =10 non-overlapping segments, each of duration 1 sec. We compute the FFT for each segment, and form the $S_k(f)$ for each of the k^{th} segments following Eq.(1) but with now $N = 1600$. The plot in Fig. 3 shows the average $\frac{1}{K}\sum S_k$. The value $\nu = 2K$ is the number of *degrees of freedom* involved the spectral estimate (Percival and Walden). Note: this relation for ν applies only to the case where there is no over-lap in the segments, and where there is no ″windowing″ of the data–it is the most simplified case, but reliable, and good place to start. Regardless, this basic rule applies: Increasing ν yields a smoother estimate of $S(f)$ with lower variance, but at the cost of lower frequency resolution. This is an important trade off you often have to consider. For our simple case, the spectral variance is

$$\mathrm{Var} S(f) \approx \frac{2S^2(f)}{\nu} \tag{3}$$

meaning the standard deviation of the spectrum at any discrete frequency f is approximately $\frac{S(f)}{\sqrt{K}}$. This effect can best be seen in Fig. 3, over the high end of the frequency range between about 600 and 800 Hz, where values of $S(f)$ are fluctuating but are also somewhat stationary. The standard deviation of such values based on the $\nu = 2$ case is a factor of $\sqrt{10}$ greater than corresponding standard deviation of values based on the $\nu = 20$ spectral representation, which we anticipate from Eq.(3).

An example of segmenting, or subsampling the data, into non-overlapping segments, and overlapping segments, each of duration 1 sec is shown in Fig. 4. For upper plot, two of the 1-sec segments are shown that form the $K = 10$ averaging segments. Another strategy is to overlap the segments, typically by 50%, as shown on the right. For the upper case example, the *spectral window* is a rectangular or uniform window. Alternatively, Fig. 4 (lower) shows the effect of breaking up the time series into 10 non-overlapping segments, and applying a Hanning spectral window to each segment which forces each data segement to start and finish at 0, and thus improves frequency resolution over that of the rectangular windw. References such as Percival and Walden, including many others, can be consulted on to learn more on how differing windowing and overlap strategies influence frequency resolution and variance. In the next lecture we show that one *best practice* is to use both a Hanning spectral window in combination with 50% overlap.

References

D. P. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*, (Cambridge University Press, New York, 1993)

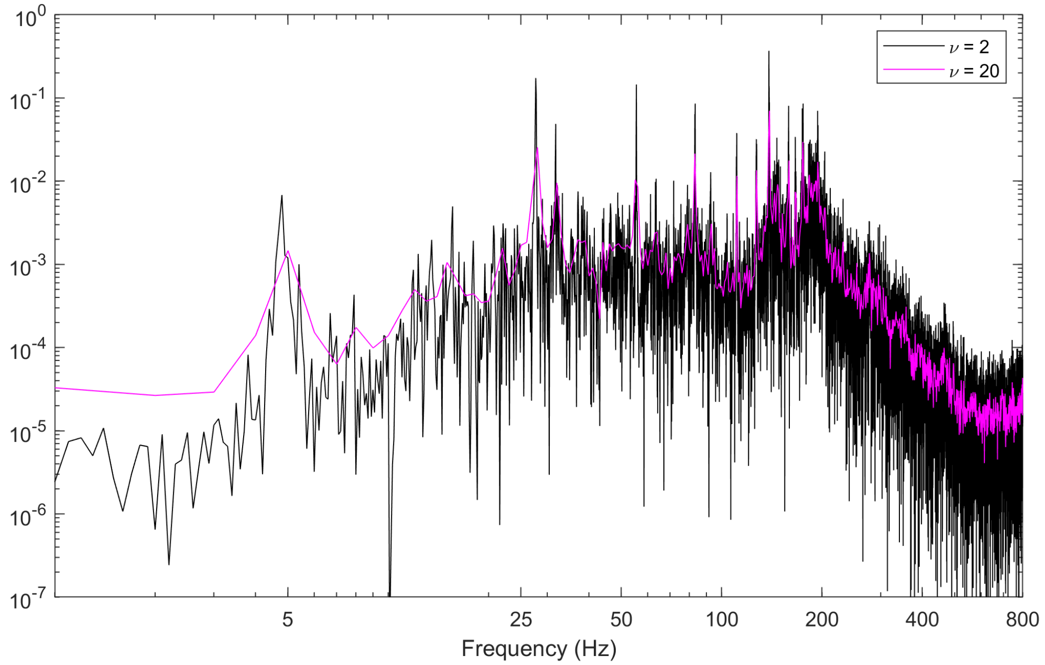

Figure 3: Proper normalized variance spectrum $S(f)$ corresponding to the data in shown in Fig. 1, based on two values of ν

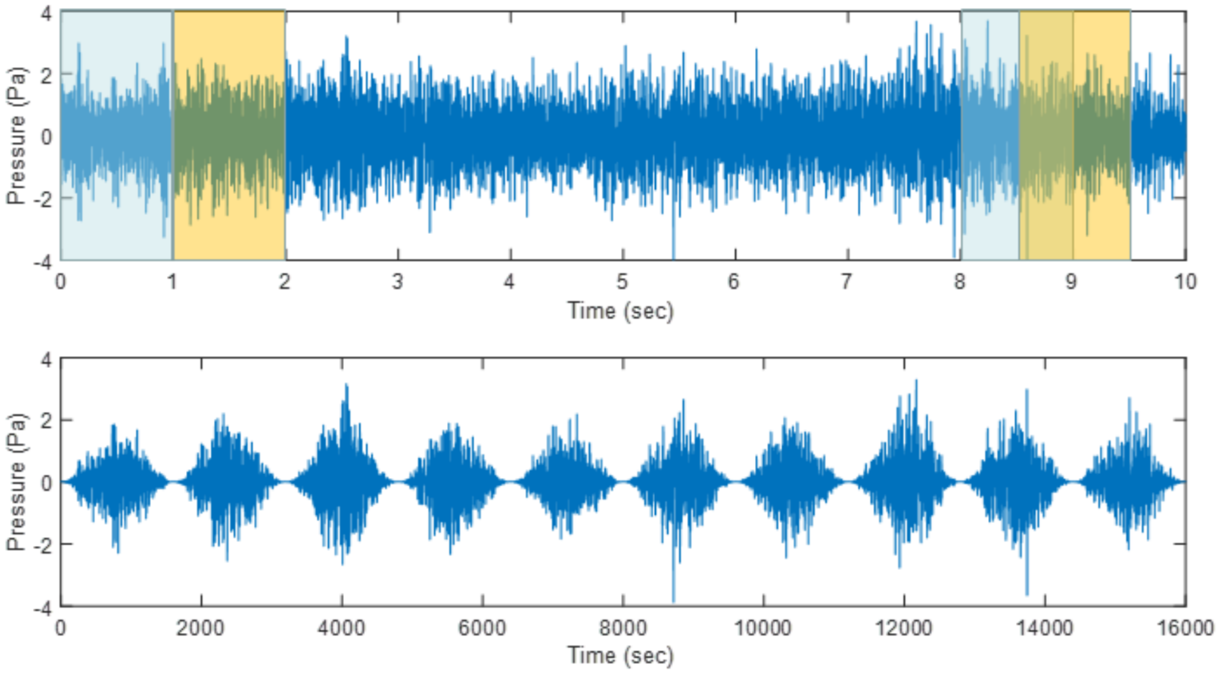

Figure 4: Upper: 16000 point time series of underwater sound corresponding to 10 sec, given sample frequency of 1600 Hz. On the left are 1-sec, non-overlapped samples, on right illustrates 50% overlapped samples. Lower: same data windowed by non-overlapping Hanning window functions.

ME525 Applied Acoustics Lecture 16, Winter 2024
Better Practices in Spectral Analysis

Peter H. Dahl, University of Washington

Tapering and Welch's overlapped segments: best practice

Previously we discussed the estimation of $S(f)$, the segmenting of data into K non-overlapping segments, or blocks, and the relation between K and degrees of freedom (DOF) ν, and that increasing ν produces a decrease in the variance of the spectral estimate. All this is true, and remains a good starting point which you can carrry out with just the FFT algorithm.

However at least one "best practice" approach is known as Welch's overlapped averaging (Percival and Walden). Figure 1 upper shows the first 48000 points of a time series (black) with two non-overlapping segments each of 24000 points. Each segment is subjected to windowing function or *taper*. The taper is made by multiplying the data by a 24000-point hanning taper, and provides some improvement in the spectral estimation (e.g., reducing what is known as spectral leakage). FFT analysis is next be taken on each segment, with the all segments averaged.

A quick note about tapering, which I've referred to in previous lectures as windowing, with both terms commonly used. When one takes an FFT sample of data, it is necessarily of finite length, say time length T. Without doing anything, this means the data have been windowed, or tapered by a boxcar function of length T, meaning the data has been multiplied by a function that equals 1 for from $t = 0$ to $t = T$ and 0 thereafter.

Multiplication in the time domain translates to convolution in the frequency domain. Thus, the true spectrum of the data, say $X(f)$ has been convolved with the spectrum of boxcar function of length T, this boxcar spectrum is $\frac{\sin 2\pi fT}{\pi T}$. However, there are more optimal tapering functions than the simple boxcar, one of which (and probably the most common) is the Hanning taper function. The effect of window data is seen in Fig. 2, which compares the spectrum of the equal-length Hanning and Boxcar functions. The upshot of this figure is that center width of the Hanning window is slightly wider and therefore gives up frequency resolution, but side lobes are reduced. In contrast the Boxcar window has slightly better frequency resolution, but with higher side lobe structure which causes spectral leakage.

There are some applications, e.g., measurement of a smaller Doppler shift in frequency, where the best frequency resolution is needed, and other applications, such as the measurement of random noise where resolution is not so essential. On balance, results from majority of applications are improved upon using the Hanning, or another optimized window, compared with the simple boxcar.

Another strategy to improve results is to overlap continuous blocks of data to reduce the vari-

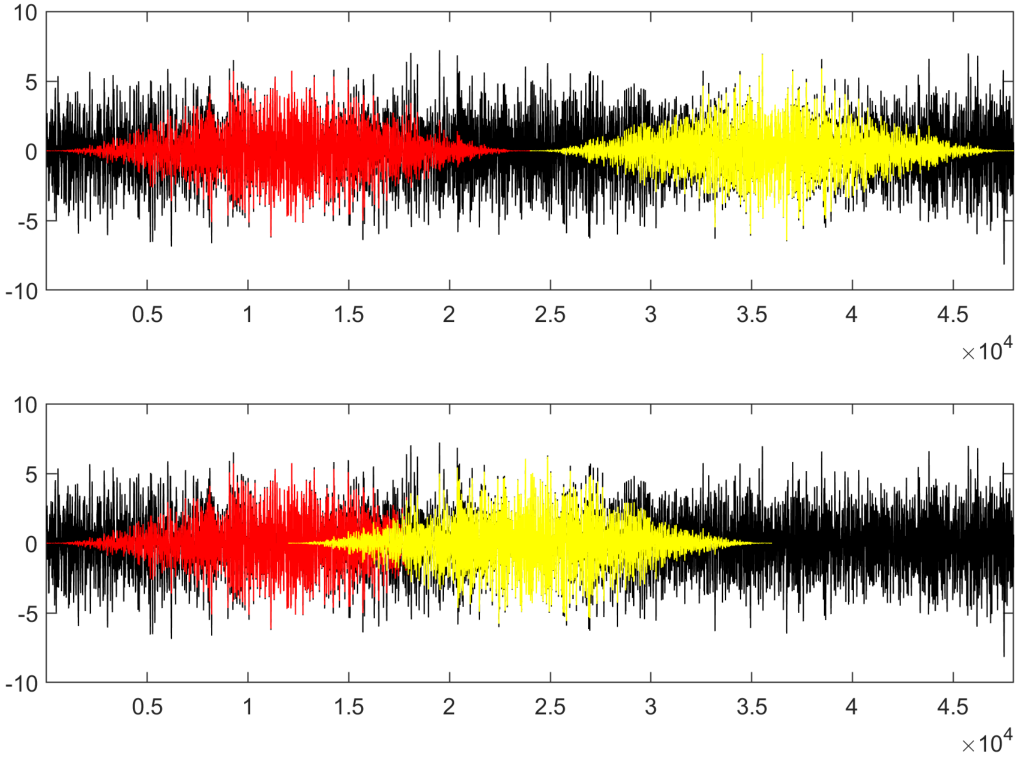

Figure 1: Upper: 48000 points of a time series (black) with two non-overlapping segments each of 24000 points, each subjected to Hanning taper function. Lower: same data 50% overlapping of the first two segments of 24000 points.

ance of the spectral estimate. The lower graph Fig. 1 shows the case where there is a 50% overlap for each block. Thus for the same sample size you can imagine there is about twice as many blocks when there is 50% overlap. Furthermore, because of block overlap some of the data lost in one block owing to the taper is regained in the adjacent block. This combination of 50% block overlap, plus a Hanning taper, is a common recommendation for spectral analysis in the engineering literature, and you should keep that in mind.

Continuing now with the case of 50% overlap. Let n be the total number of points in the time series, and n_s be the non-overlapping length (as in 24000 points in Fig. 1 upper). Then the total number of blocks is $n_b = \frac{2}{n_s}(n - n_s) + 1$. Taking a case from the start of Lecture 15, with n = 16000, and $n_s = 1600$, giving $K = 10$, then at 50% overlap n_b = 19, or about twice K as anticipated. A revised estimate for the ν is (Percival and Walden)

$$\nu \approx \frac{36 n_b^2}{(19 n_b - 1)} \tag{1}$$

Eq.(1) gives ν = 36, a good increase from $\nu = 20$ were we to use the non-overlapping segments, and the result will be a reduction in the variance of the estimate.

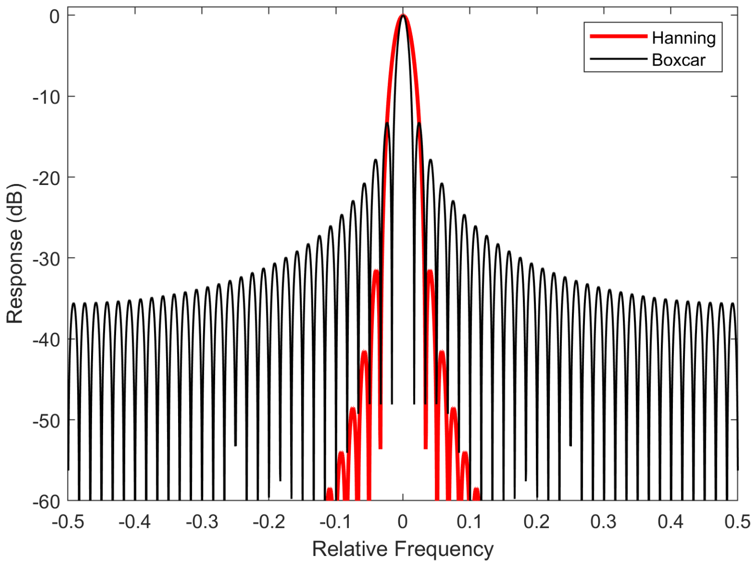

Figure 2: Magnitude spectrum of equal length Hanning and Rectangular (Boxcar) taper functions. Frequency is non-dimensional frequency. Center width of the Hanning is slightly wider, giving up frequency resolution but side lobes are reduced. Boxcar has slightly better frequency resolution, but with higher side lobe structure or spectral leakage.

Figure 3: The M/V Oregon Highway the spectrum from which was measured by a bottom hydrophone located about 100 km south of Cape Cod in March 2017. The vessel is moving about 14 knots, and these data are from the closest point of approach (CPA) from the sensor, which is about 500 m.

As specific example of these ideas we examine the variance spectrum of measurements from merchant vessel M/V Oregon Highway (Fig. 3). The variance spectrum is shown in Fig. 4 using 50 non-overlapping, non-tapered segments, (ν = 100) and then with the Welch overlap method with 50% overlap and Hanning taper (ν = 190). The result is real improvement in terms of lower variance. The Welch method is achieved with a couple of lines in matlab (provided you have signal processing toolbox)

```
[S,F] = pwelch(data_Pa,hanning(24000),12000,24000,5000);
```

This one line takes the data divided in segments of 24000 points, overlap 12000 points (50%), 24000 point hanning window, and sample frequency 5000 Hz. Using the pwelch code is quite handy because the results are automatically normalized to the variance of the original time series.

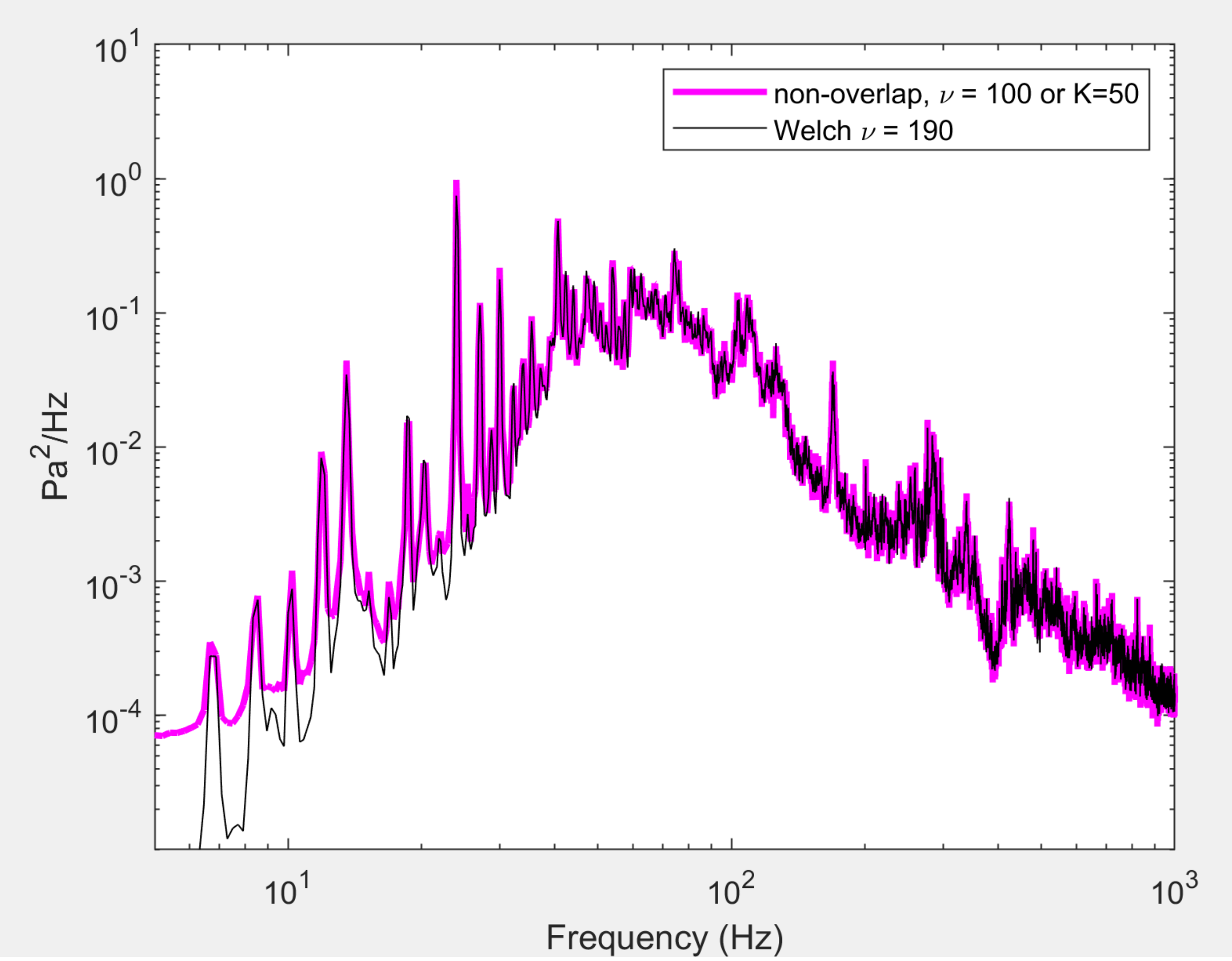

Figure 4: Spectral estimates of underwater noise measurements from the M/V Oregon Highway data based on 50 non-overlapping, non-tapered segments (ν = 100)., and based on the Welch overlap method with 50% overlap and Hanning taper (ν = 190).

Confidence intervals of the spectrum and plotting in dB

Confidence interval (CI) are commonly use to bracket a spectral estimation. For this case it is more convenient to plot $S(f)$ and the CI on a decibel scale. This can be subtle–and for sure there are many programs that do this automatically–however you need to understand what you are doing to avoid garbage.

I will distill results directly from Percival and Walden pp 256-257, and their table 256, for which we limit discussion to the so-called 90% and 95% CI, which are the most common. Also the results given here are limited to large ν of order 10 or more, which is also most common. The convention for CI is such that CI = $100(1-2p)$%, for example, a $p = 0.05$ implies a CI of 90%. We require evaluation of $Q_\nu(p)$ and $Q_\nu(1-p)$ where

$$Q_\nu(p) \approx \nu(1 - \frac{2}{9\nu} + \phi^{-1}(p)(\frac{2}{9\nu})^{1/2})^3 \tag{2}$$

From Table 256, for CI 90% find $\phi^{-1}(p) = -1.6449$ and $\phi^{-1}(1-p) = 1.6449$, for CI 95% find $\phi^{-1}(p) = -1.96$ and $\phi^{-1}(1-p) = 1.96$. Equation (2) perhaps looks a bit strange, but numbers like 1.6449 and 1.96, ought to be somewhat familiar from the study of the standardized normal probability density function. For example the probability of standardized normal random variable z such that $|z| > 1.96$ equals 0.05.

To formally plot $S(f)$ in dB along with CI compute:
$10\log_{10}\frac{\nu}{Q_\nu(1-p)} + 10\log_{10} S(f)$
for the lower bound, and
$10\log_{10}\frac{\nu}{Q_\nu(p)} + 10\log_{10} S(f)$
for the upper bound.

The result (Fig. 5) shows quite small dB-bounds (red curves) that bracket the mean estimate $10\log_{10} S(f)$ (black curve). Note: Fig. 5 is plotted as $10\log_{10} S(f)$ + 120 dB, to have the decibel values correspond to the more common underwater standard reference pressure of 1μ Pa. The expanded scale on the right-hand plot illustrates the ± dB bounds, for two peaks near 12 and 13. 5 Hz.

Repeating this but now by taking segments of length 10 times longer (Fig. 6), observe how the variance increases but the frequency resolution improves (best seen on the right expanded scale). This fundamental trade off between variance and resolution is very important to understand.

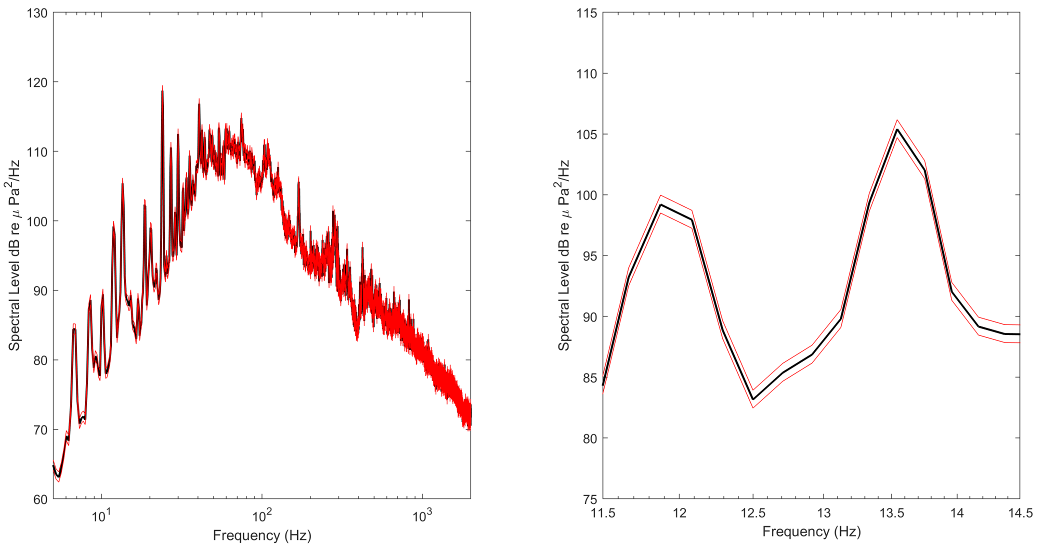

Figure 5: Spectral estimates of underwater noise measurements from M/V Oregon Highway based on Welch overlap method with 50% overlap of segments of length 24000 points, Hanning taper, giving $\nu = 190$, along with 90% confidence intervals. Data are plotted in dB re μ Pa^2/Hz. Left plot: full frequency range in log scale, right plot: expanded frequency range in linear scale

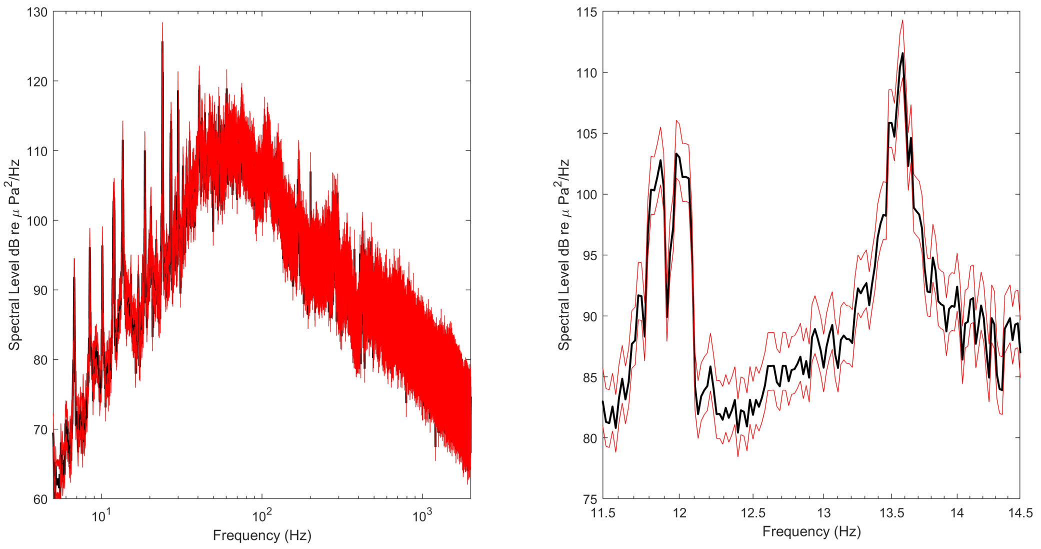

Figure 6: Spectral estimates of underwater noise measurements from M/V Oregon Highway based on Welch overlap method with 50% overlap of segments of length 240000 points, Hanning taper, giving $\nu = 19$, along with 90% confidence intervals. Data are plotted in dB re μ Pa^2/Hz. Left plot: full frequency range in log scale, right plot: expanded frequency range in linear scale

The short-time Fourier transform, used in Time-Frequency analysis

Figure 7 shows 50 sec of acoustic data from the noise emission of a jet as it was flying over water. The recording was made with buoy equipped with a microphone at height 3 m above the air-water interface. The sampling frequency is 32000 Hz, so the total number of samples N in the entire data stream is 1600000. The physics involved is like the reverse of the Lloyd mirror problem studied earlier using a real a source below the air-water interface, combined with a negative image source above the air-water interface to match the boundary condition that pressure equals zero on the boundary. Here air-to-water boundary, i.e., sound originating from above, behaves more like rigid boundary, or known as a Neumann boundary condition. For modeling, a real a source is located above the air-water interface, and a positive image source below air-water interface. The combination of the real and positive image source will satisfy this approximately rigid boundary condition.

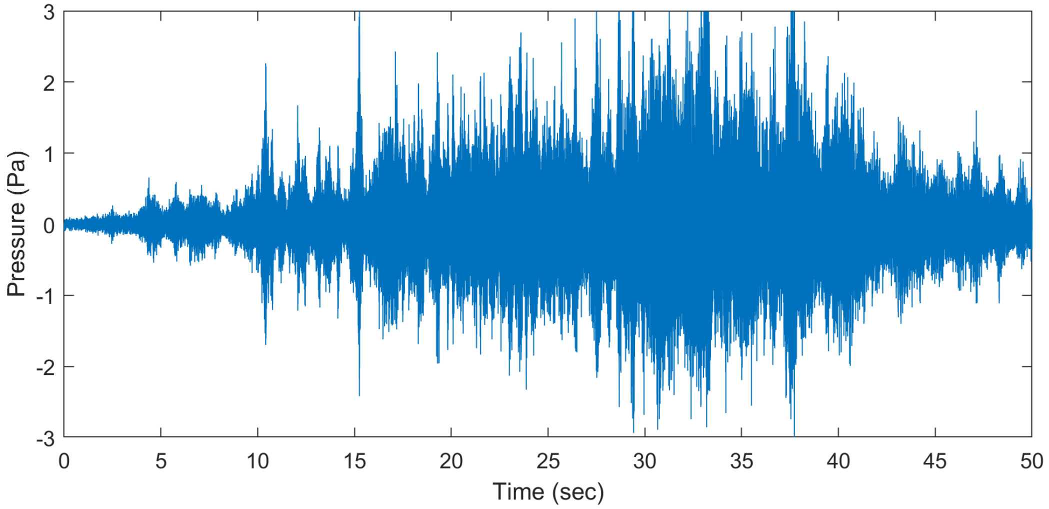

Figure 7: Jet noise recorded over water by a microphone 3 m above the air-water interface.

Figure 8 (left) shows variance spectrum $S(f)$, obtained with $K = 1$ (all data), $K = 10$ (5-sec non-overlap rectangular window), and $K = 50$ (1-sec non-overlap rectangular window), all which yield the variance of the data time series data via integration as required, and by now you should recognize the role of K or DOF ν, in determining the variance. Note: looking at the data in this manner, using $K =$ 1,10 or 50, can be somewhat problematic, as the data are not stationary insofar as the variance of the time series is changing over time, e.g., the pressure level in the middle of the data is about 20 dB greater than at the start. A more rigorous approach might be to analyze stationary segments of the data. Nevertheless, one can still proceed and recover a variance spectrum $S(f)$ and this (Fig. 8, left) is clearly informative in showing that the lion's share of the spectral energy is within frequency range 100-1000 Hz.

Figure 8 (right) reveals considerably more information about the process, and this is achieved via a short-time Fourier transform (STFT). The matlab signal processing toolbox provides several ways to do this, e.g using `STFT`, or `Spectrogram`. Shown here, however, a simple way to do this with just the FFT in case you do not have access to these toolboxes– a sort of ”poor mans” STFT.

First data vector `data_Pa` is a column vector, so its dimension is 1600000 by 1. Next reshape this into 100 column vectors of $N = 16000$ points each (or 0.5 sec), and reshaped into a matrix as follows: `data_ Pa_mat=reshape(data_Pa,length(data_Pa)/100,100);` (Be sure to do a little arithmetic check first to insure that reshaping the time series into K segments yields a matrix with the same total number of points.)

Now take the FFT of this matrix. Matlab FFT works naturally on these column vectors, and the resulting operation yields 100 FFT column vectors `X = FFT(data_Pa_mat);`. Take the magnitude squared of the complex FFT coefficients, and toss out the redundent second-half of all 100 FFT column vectors: `X = X.*conj(X)` , followed by `X = X(1:8000,:);`. The revised matrix is now 800 by 100, representing the 100 time increments of 0.5 sec. Next for completeness put `X = X*2*dt/N;`; to provide the proper variance normalization as discussed in Lecture 15. Equate X to the estimate of the STFT $S(f;t_i)$ where it is understood that t_i is a time index separated by 0.5 sec. Remember to make a 800-point frequency vector, call this `Freq`, and a good way to display this is as follows:

```
pcolor(time/2,Freq,10*log10(S))
shading('interp')
caxis([-50 -10])
ylim([0 2000])
colormap('hot')
a = colorbar;
a.Label.String = '10log_{10} S(f) ';
ylabel('Frequency (Hz)')
xlabel('Time (sec)')
```

where `time` is just an index 1:1:100. Notice I've plotted $10\log_{10}$ of $S(f;t_i)$. This is very often necessary to fully grasp the dynamic range of the spectrum.

Using this or any other approach to STFT you need to experiment as to how short an FFT to use, depending on application. Making it too short then the frequency resolution is poor, making it too long and the time resolution is poor and you can't track changes in frequency. The pattern of shown in the STFT can be understood in the context of a Lloyd mirror interference pattern between the jet plane sound source, and a positive image source below the air-water interface at depth corresponding to the height of the jet.

Another interesting example of time-frequency analysis is that of bird chirp (Fig. 9, left) from a singing robin that I recorded. This time series is very short, about 1.5 sec, and the performance of my simple approach as outlined for the jet case doesn't work as well. However, with matlab signal processing toolbox, I can use spectrogram and have the results done in one line for me. Below

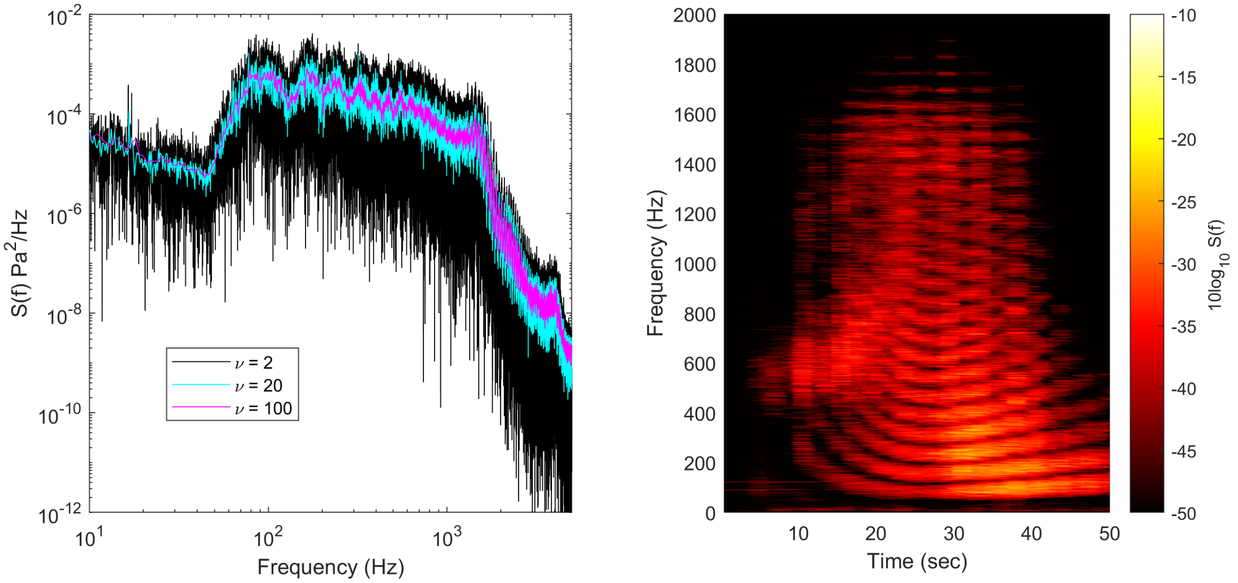

Figure 8: Left: Proper normalized variance spectra $S(f)$ corresponding to the data in shown in Fig. 2, based on three values of ν Right: A STFT of the data from Fig 2 based on 100 segments of 0.5 sec duration.

I use spectrogram on the 60000 point time series ”birdcall” with these arguments: (i) windowed segments of length 512 are made using the hanning window, (ii) overlap each window by 50% (256 points), (iii) a 512 FFT used, and (iv) identified sample frequency of 30000 Hz. Use `imagesc` with of $20 \log_{10} |S|$, and set the scale between -25 and 30 dB–one usually experiments with this. Note: spectrogram produces a complex amplitude spectrum, so need to take absolute value before the taking the $20 \log 10$ of the result. The result (Fig. 9, right) shows the interesting fine detail of the frequency modulation (and harmonics) in the bird chirp over the first 0.5 sec, seen in even greater detail in Fig. 10.

```
[S,F,T] = spectrogram(birdcall,hanning(512),256,512,30000) ;
sc=[-25 30]  %best for colormap jet
imagesc(T,F,20*log10(abs(S)),sc), axis xy, colormap (jet)
```

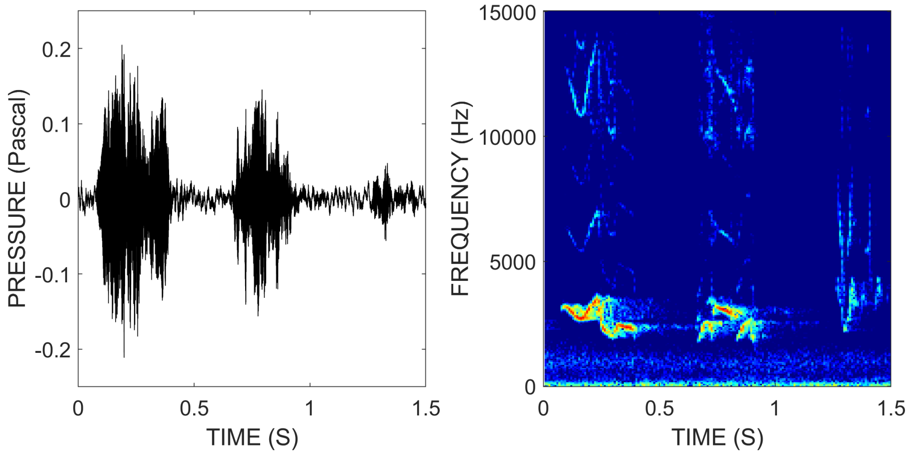

Figure 9: Left: Time series of the recorded chirp from robin sample frequency 30000 Hz, recorded at approximate range of 8 m. Right: STFT of the data using the matlab function spectrogram (see text).

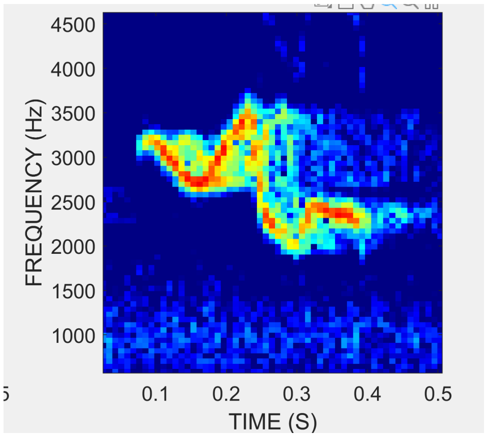

Figure 10: Expanded scale of the STFT showing frequency modulation over the 0.5 sec

References

D. P. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*, (Cambridge University Press, New York, 1993)