Additional notes for tutorial 1.

1. Convolution Theorem

The convolution of a time series x(t) with a filter h(t) to yield y(t) is given by:

$$y(t) = x * h = \int_0^T x(t-\tau)h(\tau)d\tau$$

Here h is causal and finite (up to time T). In discrete form this operation is written as:

$$y(i) = \sum_{j=1}^{N} x(i-j)h(j)$$

In Matlab convolution is done using the command *conv* or *filter*. Both conv and filter use the same time-domain algorithm but *filter* is a more general function since it can also be used for IIR filtering (infinite impulse response). IIR filters have both a feedforward (a convolution) and a feedback term.

The convolution theorem states that the Fourier Transform (FT) of a convolution is the product of the fourier transform of each component in the operation (x and h in this case):

$$FT[y] = FT[x * h] = FT[x] \cdot FT[h]$$

$$Y(f) = X(f) \cdot H(f)$$

Where we use H(f) to denote the Fourier Transform of h(t). Note that the Fourier Transforms of x, h and of their convolution y are all complex numbers.

The convolution theorem does not play a role in our signal to noise analysis but it is useful to present it here to compare it and contrast it to the correlation theorem.

2. The Correlation Theorem

The correlation between two time series x(t) and y(t) is defined as the average of the cross product of x and y for different delays τ:

$$C_{xy}(\tau) = \frac{1}{T}\int_0^T x^*(t)y(t+\tau)dt$$

Here we are averaging over the entire duration of the signals x and y: T. If the signal length's are infinite the correlation can be written mathematically as :

$$C_{xy}(\tau) = \lim_{T\to\infty}\frac{1}{T}\int_0^T x^*(t)y(t+\tau)dt$$

In discrete form the correlation is written as:

$$C_{xy}(k) = \frac{1}{N-k}\sum_{i-1}^{N-k} x^*(i)y(i+k)$$

Here k is the delay and N is the length of the time series x and y in sample points. Note that the cross-correlation is sometimes written with R instead of C (and C reserved for cross-covariance). Note that the order of the indices xy is important. As written, a positive delay, τ or k, corresponds to y coming after x. The matlab function for correlation is xcorr (and xcov for cross covariance). Note that in xcorr the convention is reverse for the order of x and y. Also the default for xcorr is to perform a cross-correlation without normalization (without dividing by N-k). This default is mostly useless and you will want to specify 'unbiased' as an option.

The correlation theorem says that:

$$S_{xy} = FT\left[C_{xy}\right] = \langle X^*(f)Y(f)\rangle$$

S_{xy} is called the cross-spectral density. The <> stand for the expectation value (or average). This average is obtained by dividing the data into segments. The optimal length of the segment depends both on the correlation time of the system (the memory of the system) and on the amount of data that you have since you will want to average over a minimum number of segments to obtain good estimates.

When x=y, the cross-correlation function becomes the auto-correlation function and the cross-spectral density becomes the power-spectral-density.

$$S_{xx} = FT[C_{xx}] = \langle X^*(f)X(f)\rangle = \langle |X(f)|^2\rangle$$

In matlab, there are many flavors of power-spectral density that are mostly different in how the signal is divided into segments. *periodogram* divides the signal into rectangular non-overlapping windows. *pwelch* divides the signal into overlapping windows where each segment is multiplied by a hamming window. *Pmtm* uses the Thompson multi-taper windows where each segment is multiplied by a series of windows (called the slepian sequences) that have different time-frequency scales and remain orthogonal to each other. The cross-spectral density can be obtained by *cpsd* (or *csd* in older versions). *Cpsd* uses the Welch method. Note that cpsd returns a complex number.

3. Signal and Noise

If the noise is additive, the noise corrupted response can be written as a sum of signal, S, and noise, N :

$$R = S + N$$

By averaging responses obtained to the same stimulus, one can obtain an estimate of the underlying signal:

$$\bar{R} \cong S$$

In single unit recordings $\bar{R}$ is the PSTH.

The noise can then be estimated by subtraction:

$$R - \bar{R} \cong N$$

The signal to noise ration can be obtained by calculating the power-spectral densities of S and R:

$$SNR = \frac{\langle S^2\rangle}{\langle N^2\rangle}$$

Where we have dropped the frequency for simplicity. To obtain a single quantifier for the overall signal to noise ratio, one can calculate the channel capacity also called the information upper bound:

$$I = \int_0^\infty log_2\left(1 + \frac{\langle S^2\rangle}{\langle N^2\rangle}\right) df \text{ bits/s}$$

The channel capacity gives the maximum information that could be transferred to this channel given that the signal and noise have Gaussian distributions. If the signal is not Gaussian, the actual maximum information that can be transmitted will be smaller.

4. The coherency and the coherence.

The coherence is used to normalize the cross-spectrum of two time series. The coherency is the cross-spectrum normalized by the squareroot of the psd. The coherency is the equivalent of the correlation coefficient for time series (*r*). The coherency is a function of frequency and is a complex number. Its inverse Fourier Transform can be taken to obtain a normalized cross-correlation function in the time domain. The coherency is:

$$\gamma_{x,y} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}$$

The coherence is the absolute value square of the coherency. The coherence is therefore a real function (of frequency) bounded between 0 and 1. The coherence can be thought of as the R^2 for time series.

$$\gamma^2 = \frac{\left|S_{xy}\right|^2}{S_{xx}S_{yy}}$$

5. The coherence and SNR.

The coherence between a single trial response (R=S+N) and the signal (S) is function of signal to noise. Using the fact that the noise is uncorrelated to the signal, it can easily be shown that:

$$\gamma_{R,S}^2 = \frac{\langle S^2\rangle}{\langle S^2\rangle + \langle N^2\rangle}$$

Equivalently:

$$\frac{\langle S^2\rangle}{\langle N^2\rangle} = \frac{\gamma^2}{1-\gamma^2}$$

The information capacity can therefore be rewritten as:

$$I_{Capacity} = \int_0^\infty log_2(1-\gamma_{R,S}^2)df$$

More generally, the coherence measures the degree of the linear relationship between two time-series. If time series y, can be obtained by convolving time series x with a filter and adding Gaussian noise the *actual* mutual information between x and y is also given by:

$$I_{Linear} = \int_0^\infty log_2(1-\gamma_{x,y}^2)df$$

6. Bias-free estimations of signal and noise.

We noted that $\bar{R} \cong S$ and $N_{est} = R - \bar{R} \cong N$. If we are using this averaging technique to estimate the signal and noise, do we obtain a bias free estimate of the SNR? To answer this question we can calculate the power-spectral densities of our estimates of signal and noise and assess whether we are under or over estimating these quantities. In the following section, I have stopped using <> to show averaging but all quantities shown are expectation values. M is the number of trials (repetitions) used to estimate the mean response. The noise across trials is uncorrelated.

The power of the average response is then:

$$|\bar{R}|^2 = |S|^2 + {|N|^2}/{M}$$

We are therefore overestimating signal power by ${|N|^2}/{M}$.

The power of our noise estimate is:

$$|N_{est}|^2 = |S - S|^2 + \left|N\left(1 - \frac{1}{M}\right)\right|^2 + \frac{M-1}{M^2}|N|^2$$

The $\left|N\left(1 - \frac{1}{M}\right)\right|^2$ term comes from the noise in the average that is the same than the noise in the trial. The $\frac{M-1}{M^2}|N|^2$ term comes from the other M-1 noise in the average that are uncorrelated with each other and with the noise in the current trial.

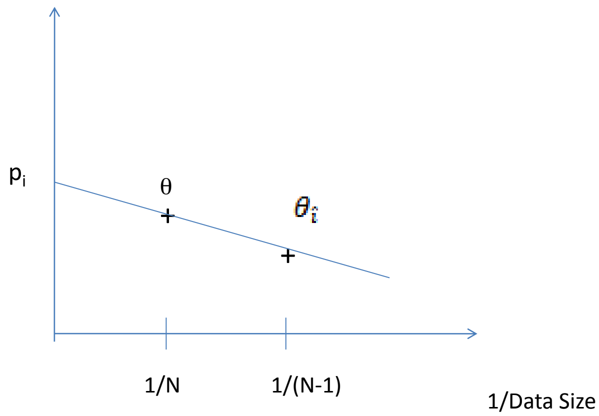
$$|N_{est}|^2 = |N|^2\left[\left(1 - \frac{1}{M}\right)^2 + \frac{M-1}{M^2}\right]$$

$$|N_{est}|^2 = |N|^2\left[\frac{M-1}{M}\right]$$

We are therefore under estimating the noise by M-1/M. To obtain bias free estimates of the SNR one could therefore correct the noise estimate by the factor M/(M-1) and then use the bias free noise estimate to correct the signal estimate by removing ${|N|^2}/{M}$. In Hsu, Borst and Theunissen, we show how to obtain bias free estimates of these signal to noise ratio using the coherence function. Our approach uses the same algebra as this derivation but it also allows one to obtain bias-free estimates of model predictions.

7. The Jackknife resampling technique.

The Jackknife is a statistical technique that enables one to generate bias free estimates of power, coherence, etc by extrapolating the results to infinite data size. Moreover, as for other resampling techniques, the Jackknife gives you an estimate of the error in your estimate. If θ is the estimate of our parameter (e.g. coherence) for all the data of size n=N and $\theta_{\hat{\imath}}$is the estimate obtained after deleting the i^{th} trial, then one can obtain a pseudo-value p_i given by a 1/n extrapolation to infinite data size.



The pseudo-value can be shown to be:

$$p_i = N\theta - (N-1)\theta_{\hat{\imath}}$$

The average value of the pseudo-values yields the bias-corrected estimator and the standard error of this mean yields an estimate of the standard error of your measurement. When the jackknife is used for coherence correction and error estimation it is often done after transformation. Thomson and Chave recommend using the inverse hyperbolic tangent.

The coherence estimator (compute_coherence_mean) in the tutorial (and in strflab) calculates the coherence using multi-taper methods that are bias corrected using the jackknife after data transformation.