Spectral Uncertainties

At the most basic level, to compute spectra, you just need to be able to compute the Fourier Transform and determine its amplitude. However, that's not quite sufficient. You'll quickly realize that an individual Fourier Transform is extremely noisy, and since its based on a single realization, it doesn't give you enough information to determine a statistical error bar. We know from computing means that the error of the mean decreases as we average more quantities together.

How do we incorporate more data into our spectra? You might imagine that you could improve your spectrum by extending the input time series from N to $2N$ data points for example. Unfortunately, although adding data points will change your spectrum, it won't reduce your noise or make the spectrum more precise at any individual frequency. Instead it will increase the number of frequencies for which you obtain results from $N/2$ to N.

Error bars for spectra depend on averaging. Our uncertainties in our spectra decrease as we average more spectra together. The challenge is to figure out how to obtain more spectra that can be averaged together. Typically what we do is to break our time series into segments, compute spectra for each of the segments, and average these to get a mean spectrum. Then we can rely on the fact that uncertainties in spectra are distributed like χ^2 to estimate the uncertainties.

If the unknown true spectrum is $f(\omega)$ and our estimate of the spectrum is $\hat{f}(\omega)$, then we can consider the ratio $\hat{f}(\omega)/f(\omega)$, where we use $\nu/2$ data segments. Formally, the probability that the estimated spectrum should be close in value to the true spectrum is:

$$P(\chi^2_{\nu,1-\alpha/2} < \nu\hat{f}(\omega)/f(\omega) < \chi^2_{\nu,\alpha/2}) = 1-\alpha \tag{1}$$

so if we want to find a 95% significance level, we set α to 0.05.

This error formulation differs from the usual error bars that we're used to seeing where we say for example that the true temperature should be the measured temperature plus or minus an uncertainty: $T = \hat{T} \pm \delta_T$. We can develop a similar expression for the true spectrum: $f(\omega)$ is in the range between $\nu\hat{f}(\omega)/\chi^2_{\nu,\alpha/2}$ and $\nu\hat{f}(\omega)/\chi^2_{\nu,1-\alpha/2}$, where ν is twice the number of segments. This expression isn't very easy to interpret, since it varies as a function of frequency, and the estimated value $\hat{f}(\omega)$ is not at the mid-point of the range.

Instead we'll make use of the ratio $\hat{f}(\omega)/f(\omega)$ which does not depend on frequency. On a log plot, error bars defined by the range between $\nu/\chi^2_{\nu,\alpha/2}$ and $\nu/\chi^2_{\nu,1-\alpha/2}$ are the same size at all frequencies, so we can easily compare spectral peaks at different frequencies.

Some statistics books include look-up tables for χ^2, but we can compute it directly in Matlab. For $N/2$ data segments, the error limits are:

```
err_low = N/chi2inv(.05/2,N);
err_high = N/chi2inv(1-.05/2,N);
```

We can plot these values as:

```
semilogy([f f],[err_low err_high]*A);
```

where we set the frequency f and the amplitude A, so that the error bar ends up positioned in a convenient spot on the plot.

Now to have $N/2$ data segments, we have to split our long data record into shorter segments. We can do this by taking M data points at a time:

```
N=length(data);
M=segment_length; % define this value
for n=1:floor(N/M)
 d=data((n-1)*M+1:n*M); %select data for the nth segment
 fd(:,n)=fft(d);        % compute fft
end
```

```
sd=sum(abs(fd(1:M/2+1,:)).^2,2)/N; % sum over all spectra (sum over 2nd index)
sd(2:end)=sd(2:end)*2;

nu=2*floor(N/M);
err_low = nu/chi2inv(.05/2,nu);
err_high =  nu/chi2inv(1-.05/2,nu);

semilogy(0:M/2,sd,[M/4 M/4],[err_low err_high]*sd(M/4))
```

There are some key details that have been neglected here. When we compute spectra, we implicitly assume that each segment perfectly repeats ad infinitum. In reality there is typically a big discontinuity between the end of the record and the beginning of a record. To prevent the low frequency information in your data from contaminating the spectra from segments, you should detrend the data. Often we demean as well and then apply a Hanning or Hamming window, which further minimizes the influence of the end members in the record.

An alternative approach to computing spectra is to compute a simple Fourier transform, and then smooth in frequency space to obtain a spectrum with error bars. The results should be essentially the same as the results from the method discussed here, though for specific applications there are good reasons why you might choose one method over the other. The textbook by von Storch and Zwiers discusses some of the details of this approach.