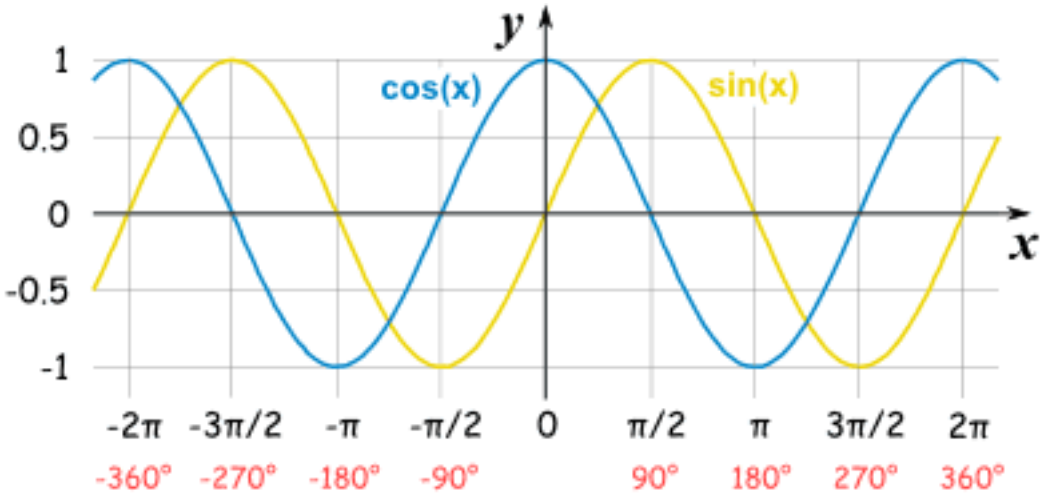
Time Series Analysis

Contents

1.1 Harmonic Analysis

1.1.1 Brief review of sines and cosines



The sine function can be written in the following way:

$$y(t) = A\sin(2\pi ft + \phi) = A\sin(\omega t + \phi) \tag{1}$$

where

- A is the amplitude, the deviation of the curve from zero
- f is the frequency in units of the number of oscillations that occur each time step
- $\omega = 2\pi f$ is the angular frequency, or the frequency in units of radians per time step
- ϕ is the phase (in radians), or the shift. When $\phi \neq 0$, the wave appears to be shifted in time by ϕ/ω time steps

As a reminder:

- record length T: length of the record such that $0 \leqslant t \leqslant T$
- wavenumber k: an integer, number of oscillations that fit inside the domain
- frequency f: number of oscillations per time step, $f = k/T$
- angular frequency ω: frequency in units of radians per time step, $\omega = 2\pi f = 2\pi k/T$
- period: number of time steps per oscillation, $1/f$

1.1.2 Fourier Sums

Basic Idea
Interpret a time or space series as a summation of contributions from harmonic functions, each with a unique temporal or spatial scale. These harmonic functions form a basis.

This is analogous to multiple regression, where you try to describe one variable as a combination of others. But, in the case of harmonic analysis, the functions to be fit (the x's) are sines and cosines:

$$\cos\left(2\pi k\frac{t}{T}\right) \tag{2}$$

$$\sin\left(2\pi k\frac{t}{T}\right) \tag{3}$$

$$\tag{4}$$

where k is an integer from 1 to N/2 (see below for definition of N), and is often called the *wave number*.

Note that in this case, $k/T = f$ from the earlier equation, that is, f is the number of oscillations divided by the total amount of time (so the number of oscillations per time step)

In this case, $y(t)$ can be decomposed as

$$y(t) = A_0 + \sum_{k=1}^{N/2} A_k \cos\left(2\pi k\frac{t}{T}\right) + \sum_{k=1}^{N/2} B_k \sin\left(2\pi k\frac{t}{T}\right) \tag{5}$$

where

- y is resolved on the time interval from $0 \leqslant t \leqslant T$
- the length of the time series is $N + 1$ (note that N needs to be even so that the highest wavenumber is an integer, but the record length needs to be odd so that all the harmonic functions fit an integer number of times into the full record
- A_k and B_k are the regression coefficients for each predictor
- each predictor is a harmonic function with frequency $2\pi k/T$ and thus fits into the interval from $t_1 = 0$ to $t_{N+1} = T$ an integral k number of times
- if $k = 1$, the function has frequency 2π radians in the full length of the data. This is the lowest resolved frequency.
- if $k = N/2$, the function has frequency 2π radians in 2 time steps. This is the highest possible frequency that can be resolved, called the Nyquist frequency
- note that if $k = N/2$, $B_k = 0$ (try drawing a picture, you can get a cosine to work, but not a sine) **Why is the limit 2π radians in 2 time steps and not in one time step? Drawing a picture helps.**

– for a given k, you are fitting 1 cosine wave and 1 sine wave with variable amplitudes into the interval from 0 to T. This is the same as fitting 1 cosine wave and no sine wave, with variable amplitude and variable phase into the same interval. In both cases, you get two predictors for each k.

1.1.3 More on the Nyquist frequency and aliasing

The Nyquist frequency is the highest frequency that can be resolved in your data:

– one oscillation per 2 time steps (1 oscillation per 2 Δt)

– wavenumber k = (length of time series - 1)/2

What if you have higher frequencies in your data? In this case, they will get *aliased* onto lower frequencies. This is a problem when there is a large amount of variance in frequencies higher than the Nyquist frequency. This is a very important issue to be aware of.

Note that the Nyquist frequency is completely dependent on your data time step (temporal resolution of your data)...*NOT the total amount of data you have.* The smaller the time step, the higher the Nyquist frequency.

Example: PYTHON NOTEBOOK: HARMONIC_ANALYSIS_NYQUIST

1.1.4 Discrete Fourier Transform

In the case of *discrete, evenly spaced data...*

Consider the domain $0 \leqslant t \leqslant T$ where 0 and T coincide with data points 1 and $N + 1$, and N is an even number. Then

1. the functions are of the form

$$\cos\left(2\pi k \frac{t}{T}\right) \tag{6}$$

$$\sin\left(2\pi k \frac{t}{T}\right) \tag{7}$$

$$\tag{8}$$

2. the average of each sine/cosine function on the interval from $t_1 = 0$ to $t_{N+1} = T$ is equal to 0, so, $A_0 = \overline{y}$

3. the harmonic functions (the predictors) are mutually orthogonal on the interval, for example,

$$r(\cos x, \sin x) = 0 \tag{9}$$

$$r(\cos x, \cos 2x) = 0 \tag{10}$$

4. the regression coefficients A_k of $y(t)$ onto the harmonic function $x_k(t)$ can be written as

$$A_k = \frac{\overline{x_k' y'}}{\overline{x_k'^2}} \tag{11}$$

which follows from the normal equations for multiple regression.

$$\begin{bmatrix} \overline{x_1^2} & \overline{x_1 x_2} & \overline{x_1 x_3} & \dots \\ \overline{x_2 x_1} & \overline{x_2^2} & \overline{x_2 x_3} & \dots \\ \overline{x_3 x_1} & \overline{x_3 x_2} & \overline{x_3^2} & \dots \\ \dots & & & \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \dots \end{bmatrix} = \begin{bmatrix} \overline{x_1 y} \\ \overline{x_2 y} \\ \overline{x_3 y} \\ \dots \end{bmatrix}$$

Here, since the predictors are orthogonal, the covariance between the cross terms are zero, and thus, we get the above equation for the A_ks

Recall that in the case of linear regression, $y = a_0 + a_1 x$, $a_1 = (\overline{x' y'})/\overline{x'^2}$ and $a_0 = \overline{y} - a_1 \overline{x}$

5. the harmonic functions each have variance

$$\overline{x_k'^2} = \frac{1}{2} \tag{12}$$

except for $k = N/2$, when the variance of the cosine term is 1 and the variance of the sine term is 0.

6. From the above, the solutions for each predictor $A_k, B_k = \frac{\overline{(x_k' y')}}{\overline{x_k'^2}}$ reduce to:

$$\begin{aligned} A_k &= 2 \cdot \overline{\cos\left(2\pi k \frac{t}{T}\right) \cdot y'(t)} && (13) \\ B_k &= 2 \cdot \overline{\sin\left(2\pi k \frac{t}{T}\right) \cdot y'(t)} && (14) \\ A_{N/2} &= \overline{\cos\left(2\pi k \frac{t}{T}\right) \cdot y'(t)} && (15) \\ B_{N/2} &= 0 && (16) \end{aligned}$$

where the 2 in front of the coefficients comes from the fact that $\overline{x_k'^2} = \frac{1}{2}$

7. thus, $y(t)$ can be written as

$$y(t) = \overline{y} + \sum_{k=1}^{N/2-1} \left(A_k \cos 2\pi k \frac{t}{T} + B_k \sin 2\pi k \frac{t}{T} \right) + A_{N/2} \cos \pi N \frac{t}{T} \tag{17}$$

Example: NOTEBOOK EXAMPLE_HARMONIC_COEFF_CALCULATOR.IPYNB

These results can be converted into amplitude-phase form (using just cosine instead of sine and cosines) to obtain

$$y(t) = \overline{y} + \sum_{k=1}^{N/2} C_k \cos\left(2\pi k \frac{t}{T} - \phi_k\right) \tag{18}$$

where

$$C_k^2 = A_k^2 + B_k^2 \tag{19}$$

This follows from the pythagorean theorem, with A_k along the x-axis and B_k along the y and the angle ϕ representing the angle between the two.

The fraction of variance explained by a particular harmonic function is given by

$$R^2(y, x_k) = \frac{\left(\overline{x_k' y'}\right)^2}{\overline{x_k'^2} \cdot \overline{y'^2}} \tag{20}$$

You've seen this before in Linear Regression. Using the regression coefficient equations for A_k and B_k, this is equivalent to

$$R^2 = \frac{A_k^2}{2\overline{y'^2}} \text{ for the cosine functions} \tag{21}$$

$$R^2 = \frac{B_k^2}{2\overline{y'^2}} \text{ for the sine functions} \tag{22}$$

In the special case of $k = N/2$,

$$R^2 = \frac{A_{N/2}^2}{\overline{y'^2}} \text{ for the cosine function} \tag{23}$$

The *actual*, not fraction, of the variance explained by a particular harmonic k is thus,

$$\text{variance explained } = \frac{A_k^2 + B_k^2}{2} = \frac{C_k^2}{2} \tag{24}$$

Example: SHOW 12_SPECTRAL_ANALYSIS_MOTIVATION.PDF

1.2 The power spectrum

The plot of $C_k^2/2$ versus k is referred to as the spectrum of $y(t)$

- if t represents time, it is called the *frequency spectrum*
- if t represents space, it is called the *wavenumber spectrum*

The plot of $C_k^2/2$ versus k is a line spectrum: it is defined only for integral values of k. While the line spectrum is fine if we have an infinite sample, it has serious drawbacks if we only have a finite data record:

1. integral values of k do not have any special relationship to the population being sampled, and are completely dependent on the length of the data record T which is presumably arbitrary

2. the individual spectral lines each contain about 2 degrees of freedom: $N/2$ estimates from N total data points

3. except for diurnal/seasonal cycles, most geophysical phenomena are only quasi-periodic, and thus, do not conform to the idea of a single integer wavenumber, but are likely better represented as spectral bands of finite width

The above considerations suggest the notion of a continuous power spectrum, where the variane $y_(t)$ is given per unit frequency (or wavenumber):

$$\overline{y'^2} = \int_0^{k^*} \Phi(k)dk \tag{25}$$

where

- $\Phi(k)$ is the continuous power spectrum
- k^* corresponds to the Nyquist frequency, i.e. one cycle per $2\Delta t$ where Δt is the constant spacing between data points

Note that $\Phi(k)$ is simply a continuous version of $C_k^2/2$ and is estimated either by smoothing the line spectrum or by averaging many line spectra together.

Two ways of obtaining more degrees of freedom:

1. **Average adjacent spectral estimates together**: lose resolution, but gain dof. For example, if you average 10 adjacent spectral estimates together, the frequency resolution decreases by a factor of 10 but the number of dof increases by a factor of 10

2. **Average realizations of the spectra together**: lose information at lower frequencies but gain dof. For example, if you subdivide the data into 10 time series of equal length, the lowest frequencies resolved increases by a factor of 10, but the number of dof also increases by a factor of 10

There is a trade-off here between resolution and reproducibility (degrees of freedom) that is always lurking in the background. You cannot get away from this one. It is somewhat akin to Heisenberg's Uncertainty Principle.

High Resolution/High Information/Low Quality Statistics
$\Downarrow$
Lower Resolution/Lower Information/Higher Quality Statistics

The degrees of freedom per spectral estimate is given by

$$\mathsf{D.O.F.} = \mathsf{N}/\mathsf{M}^* \tag{26}$$

where M^* is the number of independent spectral estimates and N is the number of actual data points.

So, for line estimates, $\mathsf{M}^* = \mathsf{N}/2$, and so, $\mathsf{D.O.F.} = \mathsf{N}/(\mathsf{N}/2) = 2$

Note: as long as we use a red-noise fit to the data as our null hypothesis, we don't need to reduce N to account for autocorrelation.

Example: NOTEBOOK SPECTRAL_DOF.IPYNB

1.2.1 Methods for computing the power spectrum

Method 1: The direct method

Calculate C_k through harmonic analysis (i.e. as per the analytic solutions for the discrete Fourier transform to get A_k and B_k).

- it is easy to code, but, not very efficient (numerous redundancies in the A_k and B_k calculations
- the Fast Fourier Transform (FFT) is a widely available routine and takes advantage of these redundancies
- transform assumes cyclic continuity (waves fit fully into the domain), it is typical to *taper the edges* of the time series (more on windowing later)

Method 2: The lag correlation method

This method is less often used, but it is still good to know about. It is based on a theorem by Norbert Weiner, namely, that

> *the autocorrelation function and the power spectrum are Fourier transforms of each other* (see Hartmann's notes for details)

Hence, the power spectrum can be found by performing harmonic analysis on the autocorrelation function within the interval $-\mathsf{T}_\mathsf{L} \leqslant \tau \leqslant \mathsf{T}_\mathsf{L}$, where T_L is the maximum time lag.

Example: SLIDES SHOWING AUTOCORRELATION, POWER SPECTRUM PAIRS

1.2.2 Plotting the Power Spectrum

There are two basic schemes for plotting the power spectra

1. the linear scale
2. the logarithmic scale

In the linear scale, you plot the frequency $f/\omega/k/T$ versus the power spectral density $\Phi(\omega)$. In this case,

$$\int_{\omega_1}^{\omega_2} \Phi(\omega) d\omega \tag{27}$$

is the total variance in the data between ω_1 and ω_2.

When the frequency interval of interest ranges over several orders of magnitude, it is useful to plot the natural log of the x-axis (e.g. frequency). In this case, the y-axis scale is set to $\omega \cdot \Phi(\omega)$ such that the visual area under the curve remains proportional to the total variance, i.e.

$$\int_{\ln \omega_1}^{\ln \omega_2} \omega \cdot \Phi(\omega) d(\ln \omega) = \int_{\omega_1}^{\omega_2} \Phi(\omega) d\omega \tag{28}$$

since $dx = x \cdot d(\ln x)$ since $d(\ln x) = 1/x \cdot dx$.

In the log scale, the low-frequency end of the spectrum is stretched (compared to the high-frequency end).

It is often useful to normalize the total area under the curve to 1, so that the area between the two frequencies is equal to the fraction of variance explained.

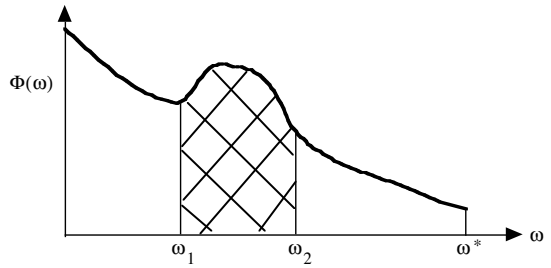


Fig. 6.11 *Schematic of a continuous power versus frequency spectrum for which the power or variance between two frequencies is proportional to the integral under the curve..*

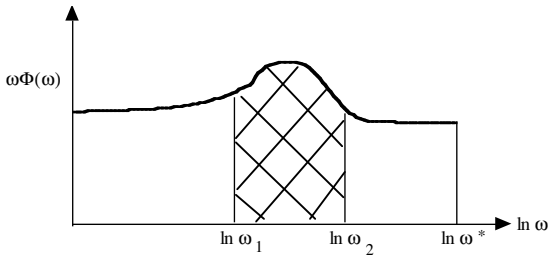


Fig. 6.12 *Schematic of a continuous frequency times power versus log of frequency spectrum for which the power or variance between two frequencies is proportional to the integral under the curve.*

figures taken from Chapter 6b of Hartmann's notes

1.3 The Complex Fourier Transform

The Discrete Fourier Transform: regression harmonic analysis of $y(t)$ to obtain C_k^2.

Until now, we have presented the Fourier transform using real arithmetic (sines and cosines),

however, it is much more efficient to write the Fourier Transform (FT) in complex arithmetic:

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \tag{29}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{30}$$

ω is the radial frequency (radians per units time), and $F(\omega)$ is generally a complex number. Note that different formulations will place the $1/(2\pi)$ in different parts of the formulas.

To see the connection between the real FT and the complex FT, it helps to recall that:

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{31}$$

$$\cos\theta = \frac{e^{i\theta}+e^{-i\theta}}{2} \tag{32}$$

$$\sin\theta = \frac{e^{i\theta}-e^{-i\theta}}{2i} \tag{33}$$

1.4 Statistical significance of spectral peaks

The statistical significance of spectral peaks are typically assessed by testing against the null hypothesis that the data is just "red noise". That is, it has low-frequency variability but is not periodic.

The amplitude of a spectral peak is tested against the spectrum of the red-noise fit to the data.

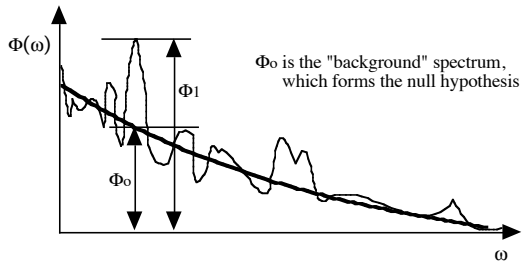


Figure. 6.18 *Schematic of a spectrum, its null hypothesis spectrum and the ratio of the variances at a frequency of interest.*

figures taken from Chapter 6b of Hartmann's notes

The significance of the ratio of variances Φ_1 to Φ_0. The significance of this ratio can be assessed using the f-statistic:

$$F = \frac{S_1^2}{S_0^2} \tag{34}$$

where S_1^2 and S_0^2 are the variances of the original spectrum and the red-noise fit. Remember - the power spectrum is this variance!

The F-statistic requires the degrees of freedom ν_1 and ν_0 for the denominator and numerator.

- for red-noise (ν_0), we typically assume it is a large number
- for the real time series (ν_1), use $N/(M_{chunk}/2) = N/(\#$ of spectral estimates)

Remember that you can only use a priori statistics if you expect a peak at a given frequency. Otherwise, you need to use a posteriori statistics, that is, you need to consider the probability that one frequency out of $M_{chunk}/2$ (the number of spectral estimates) gave you a significant peak...that is, your true significance level is

$$(0.99)^{M_{chunk}/2} \tag{35}$$

if you test each peak at 99% confidence.

Example: NOTEBOOK TESTING_SPECTRAL_SIGNIFICANCE.IPYNB

Example 4.2

F-statistic & spectral peaks

If you have a daily time series that is 512 days in length and you have split it into 128-day chunks (4 separate realizations) and calculated the spectrum. If you have a peak in your spectrum that is 3 times higher than the red noise value, is this significant at the 99% confidence level?

---

1. let's assume the peak is at a location we expected a priori
2. we need to find out whether $F_{crit} < 3$, in which case, we can reject the null hypothesis that the peak is just random fluctuations of a red noise time series
3. we need to determine the degrees of freedom for our data and the red-noise time series, ν_{data}, ν_{red}
4. $\nu_{data} = N/(M_{chunk}/2) = 512/(128/2) = 8$
5. ν_{red} = large number (typically greater than 100)
6. remember that for the F-statistic, the data is the numerator and the red-noise is the denominator
7. in this case, using our table, the 99% level $F_{crit} = 2.69$
8. since $F_{crit} = 2.69 < 3$, our peak passes the test and we can reject the null hypothesis

1.4.1 Finding the power spectrum of red noise

Recall that the autocorrelation function for red-noise is

$$\gamma(\tau) = \exp\left(-\frac{\tau}{T_e}\right), \tag{36}$$

where T_e is the e-folding time-scale.

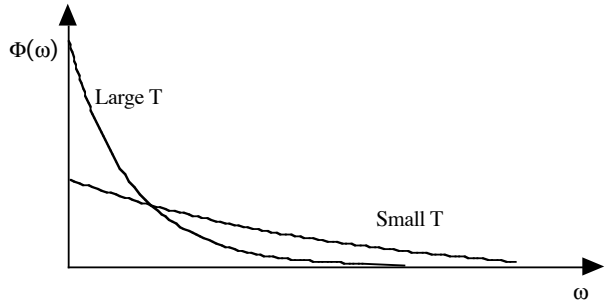
The power spectrum is the FT of $\gamma(\tau)$,

$$\Phi(\omega) = \int_{-\infty}^{\infty} e^{-\tau/T_e} e^{-i\omega\tau} d\tau \tag{37}$$

After integration,

$$\Phi(\omega) = \frac{2T_e}{1 + T_e^2\omega^2} \tag{38}$$

where T_e is the e-folding timescale of the real data and ω is the radial frequency.



figures taken from Chapter 6b of Hartmann's notes

- it is typical to set the area under the spectrum to either 1 or the total variance of the data you are fitting to
- the amplitude will appear different if you plot versus k, period, ω, etc.
- in the real world, we only have an estimate of red-noise based on a limited sample, while the above is for an infinitely long sample. Thus, in practice, use the discrete red-noise spectrum outlined in Hartmann and developed by Gliman et al. (1963) where its *unnormalized* form is given by

  $$\Phi(\omega)_{\text{discrete}} = \frac{1 - \rho^2}{1 - 2\rho\cos\left(\frac{h\pi}{N/2}\right) + \rho^2} \tag{39}$$

  where $h = [0 : 1 : N/2]$ and ρ is the autocorrelation of the red-noise process at lag 1.
- although, note that this discrete version doesn't differ much from the theoretical version, and the theoretical version is more *conservative*
- my note: FFT assumes the data is stationary - must remove any trends before analysis

Example: Show slide of comparison between theory and Gilman

1.5 Windows due to finite data

Two key problems with spectral analysis of discrete data (not a continuous time series):

1. Aliasing: we only resolve frequencies lower than the Nyquist frequency ($1/(2\Delta t)$, and frequencies higher than this get aliased to lower frequencies
2. Leakage: we are assuming that all waveforms stop and start at $t = 0$ and $t = T$, but in the real world, many of these wave numbers may not complete a full integer number of cycles throughout the domain, causing *spectral leakage* to other wave numbers

1.5.1 Convolution & Convolution Theorem

The leakage issue (finite domain) is handled by adding a window to the data. To understand these windows, one must first understand convolution.

The convolution of two functions G and g is defined as:

$$(w * g)(t) \stackrel{def}{=} \int_{-\infty}^{\infty} w(\tau) g(t+\tau) d\tau \tag{40}$$

Let's do an example to see what this equation is actually telling us (turns out it isn't that complicated!)

- let $w(\tau)$ = [0 1/3 1/3 1/3 0] for τ = -2, -1, 0, 1, 2
- let $g(t)$ = [9 7 8 6 5 4 3 8 9 10 11] for t = 1, 2, 3, 4...
- $(w * g)(t = 4) = 7 \cdot 0 + 8 \cdot 1/3 + 6 \cdot 1/3 + 5 \cdot 1/3 + 4 \cdot 0 = 19/3$
- note that in the above example, $-2 \leqslant \tau \leqslant 2$

Thus, the convolution is really just a sliding window, or, a sliding weighted average!

Warning: Software packages may not allow τ to be negative, and thus, w is not centered. Read the fine print!

As a reminder:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{-i\omega t} d\omega \tag{41}$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \tag{42}$$

Convolution Theorem

$$\mathcal{F}(f(t)g(t)) = \mathcal{F}(f(t)) * \mathcal{F}(g(t)) = F(\omega) * G(\omega) \quad (43)$$
$$\mathcal{F}(f(t) * g(t)) = \mathcal{F}(f(t)) \cdot \mathcal{F}(g(t)) = F(\omega) \cdot G(\omega) \quad (44)$$

where $\mathcal{F}$ is the Fourier Transform operator and capital letters denote the Fourier Transform of their lower-case functions.

In English, the convolution theorem tells us that the Fourier Transform of the product of two functions is the convolution of their individual Fourier Transforms (and vice versa).

Now, consider what this means for a time series of finite length, where we are in essence applying a windowing function $g(t)$ to our data, so that we can only see the time intervals from $0 \leqslant t \leqslant T$ rather than $-\infty \leqslant t \leqslant \infty$.

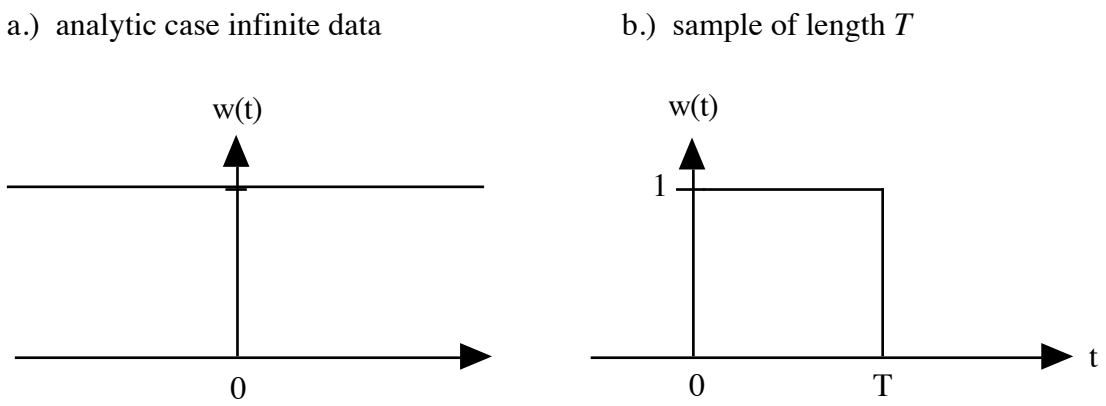


Fig. 6.13 *Schematic comparing the window function of an infinite time series, with that of a finite time series..*

Infinite window: $g(t) = 1$ for all t.
Finite window: $g(t) = 1$ for $0 \leqslant t \leqslant T$ and $g(t) = 0$ elsewhere.

Thus, when we think we are calculating the Fourier transform for our finite data $z(t)$, we are actually calculating the Fourier transform of $z(t) = f(t) \cdot g(t)$, (where $f(t)$ is our data with an infinite window) thus,

$$\mathcal{F}(f(t)g(t)) = \int_{-\infty}^{\infty} f(t) \cdot g(t) e^{-i\omega t} d\omega \quad (45)$$

From the convolution theorem, we know that

$$\mathcal{F}(f(t)g(t)) = \mathcal{F}(f(t)) * \mathcal{F}(g(t)) = F(\omega) * G(\omega) \quad (46)$$

$$F(\omega) * G(\omega) = \int_{-\infty}^{\infty} F(\omega_0) G(\omega - \omega_0) d\omega_0 \quad (47)$$

So, the Fourier Transform of our discrete data $f(t)$ with window $g(t)$ can be found by convolving the individual Fourier Transforms $F(\omega)$ and $G(\omega)$!

One way to think about this is that our infinite Fourier Transform $F(\omega)$ gets smoothed out by a moving weighted average that looks like the Fourier Transform of $g(t)$ (namely, $G(\omega)$).

1.5.2 Response function for various windows $g(t)$

The function $G(t-\tau)$ is called the *response function* of window $g(t)$ that acts on the original time series (where $g(t)$ is a window in time, not in frequency).

Ideal Case
In the ideal world, we would like our window $g(t)$ to give us a response function of $G(\omega-\omega_0)$ such that

$$G(\omega-\omega_0) = 1, \text{ when } \omega = \omega_0 \tag{48}$$
$$G(\omega-\omega_0) = 0, \text{ when } \omega \neq \omega_0 \tag{49}$$

That is, a delta function at lag 0 (or, where $\omega - \omega_0$).

Boxcar Window
In reality, the response function for our discrete window is very different from this, for example, consider the naive window that we all automatically use when analyzing data of finite length, i.e. the Boxcar Window:

$$g(t) = 1 \text{ for } 0 \leqslant t \leqslant T \tag{50}$$
$$g(t) = 0 \text{ for all other t} \tag{51}$$
$$\tag{52}$$

The Fourier Transform of the Boxcar Function is:

$$G(\omega) = \frac{\sin(\omega T/(2))}{\omega T/(2)} \tag{53}$$
$$= 2\mathrm{sinc}(\omega T/(2\pi)) \tag{54}$$

Where the normalized sinc function ("sine cardinal") used in digital processing is defined as

$$\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{55}$$

Example: SLIDE OF BOXCAR RESPONSE FUNCTION

Example: NOTEBOOK CONVOLUTION_EXAMPLE.IPYNB

- **smoothing:** when the spectrum is calculated through a boxcar, the window acts to smooth and distort the true spectrum

- **side lobes:** note the *side lobes*, these can be a real problem as they add frequencies to your spectrum that may not have been present in your original data - the reason for the side lobes is that these are the higher frequency harmonics needed to try and fit the sharp corners of the boxcar window
- **small T:** note that for small T, the frequency window (response function) is much wider, and thus, there is more smoothing among the different frequencies
- **large T:** for large T, the frequency window (response function) is more like a delta function, and becomes closer and closer to the ideal response function
- for large T, this makes sense, because the edges of your "boxcar" are just a tiny part of your data, while for small T, the edges may actually be close to a lot of your data since you don't have much

Hanning Window (Hann Window, Cosine Bell or Tukey Window)

The Hanning Window is one of many windows that acts to taper the end of the time series, rather than have a sharp rectangular corner. It is automatically applied in Matlab spectral routines like *spectrum.m* and *spectrum.welch*.

$$w(\mathsf{t}) = \frac{1}{2}\left(1 - \cos\frac{2\pi\mathsf{t}}{\mathsf{T}}\right) \text{ for } 0 \leqslant \mathsf{t} \leqslant \mathsf{T} \tag{56}$$

$$= \frac{2}{\mathsf{T}}\cos^2\frac{\pi\mathsf{t}}{\mathsf{T}} \tag{57}$$

The response function for the Hanning window is

$$\mathsf{G}(\omega) = \mathtt{sinc}(\omega\mathsf{T}/(2\pi)) + \frac{1}{2}\left(\mathtt{sinc}(\omega\mathsf{T}/(2\pi)+1) + \mathtt{sinc}(\omega\mathsf{T}/(2\pi)-1)\right) \tag{58}$$

The first term is identical to the boxcar and the last two terms cancel the side lobes.

Example: NOTEBOOK BOXCAR_HANNING_RESPONSE_FUNCTIONS.IPYNB

Note that once again, you cannot get something for nothing however, as the central lobe widens in order to remove the side lobes. However, the advantages of removing the side lobes most often outweighs the disadvantage of the extra smoothing.

Note that this additional smoothing isn't considered in your DOF calculation, and thus, there is an additional factor added into the DOF calculation to account for the smoothing by the window:

$$\mathsf{n} = \frac{\mathsf{N}}{\mathsf{M}_{\mathtt{chunk}}/2} \cdot \mathsf{f}_w \tag{59}$$

where f_w is a factor to compensate for the extra smoothing by the window. It is generally between 1 and 1.5 depending on how much smoothing the window does. For a Hanning window, $\mathsf{f}_w = 1.2$.

Hamming Window

This window is similar to the Hanning window, but with a slight modification, providing a slightly more optimal reduction in the side lobes and slightly more smoothing of the central lobes.

1.5.3 Applying overlaps of the windows

If you have a time series that is 400 data points, and you split it into 4 chunks of 100 points each to increase your degrees of freedom, you will end up using very little of the data at the edges of your chunks, due to the tapered window. But, this data is no more important than the data in the center of the chunks!

To treat each data point more equally, often one will apply the window to one chunk of data, and then include a 50% overlap with that data when calculating the second chunk of data. In this case, you will have more than 4 chunks (you will have 7 chunks). It turns out that even though it may feel like you are double counting data, you actually gain 2 degrees of freedom per chunk (including those overlapping) if you are using a Hanning window. That is, if you use a 50% overlap with a Hanning window you will have $7 \times 2 \times 1.2 = 16.2$ degrees of freedom per spectral estimate. The reason you can still consider each overlapping chunk as independent is because you are able to take advantage of correlations you wouldn't have otherwise (recalling that the spectrum is the Fourier transform of the autocorrelation).

This methodology is sometimes given the acronym:
WOSA: Welch's Overlapping Segment Analysis.

Example: DRAW A PICTURE TO SHOW HOW THIS WORKS

Worked Example: WELCH'S 1967 PAPER

1.6 Filtering

While we typically think of *windowing* our data in the time domain, windowing in the frequency (or spectral) domain is often referred to as *frequency filtering*, but we can also filter in the time domain (e.g. moving average). More on these later.

- low pass filter: isolates low frequencies
- high pass filter: isolates high frequencies
- bandpass filter: removes both high and low frequencies and isolates frequencies in a band in the middle of the frequency domain

Example: DRAW EXAMPLE USING A BOXCAR WINDOW IN SPECTRAL SPACE

WARNING: always be careful when frequency filtering. A bandpass filter will make even noise look periodic and interesting.

1.6.1 General idea of frequency filtering

Recall harmonic analysis in amplitude/phase form:

$$y(t) = \overline{y} + \sum_{k=1}^{N/2} C_k \cos\left(2\pi k \frac{t}{T} - \phi_k\right) \tag{60}$$

The filtered time series $y(t)_{filtered}$ can be written as:

$$y(t)_{filtered} = \overline{y} + \sum_{k=1}^{N/2} R(k) C_k \cos\left(2\pi k \frac{t}{T} - \phi_k\right) \tag{61}$$

where $R(k)$ is the response function of the filter and is given by:

$$R(k) = \frac{\text{amplitude of output time series at wavenumber k}}{\text{amplitude of input time series at wavenumber k}} = \frac{C(k)_{filtered}}{C(k)_{unfiltered}} \tag{62}$$

Generally, $R(k)$ is given as a function of frequency $R(\omega)$, not wavenumber, but these are just units.

In general, $R(\omega)$ may be complex, since a filter can change both the amplitude and the phase for a given frequency. For the most part though, you are interested in the amplitude, hence, we will only consider the real case.

Assuming $R(\omega)$ is not complex and thus doesn't impact the phase of the spectrum:

$$R(\omega) = \frac{C(\omega)_{filtered}}{C(\omega)_{unfiltered}} \tag{63}$$

$$|R(\omega)|^2 = \frac{\Phi(\omega)_{filtered}}{\Phi(\omega)_{unfiltered}} \tag{64}$$

There are two general methods for filtering a time series:

1. in the frequency domain
2. in the time domain

1.6.2 Filtering in the frequency domain

For the frequency domain, you simply transform the time series into the frequency domain (e.g. FFT), multiply $C(\omega)_{original}$ by your desired $R(\omega)$, and then reconstruct the data back into the time domain using $C(\omega)_{filtered}$.

This is a very effective way of filtering, and ensures you keep the frequencies that you want. However, like any filter, it can introduce spurious behavior in the time domain.

For example, note that choosing a single spectral estimate, say, $k = 3$, is in essence applying a boxcar window in the frequency domain with a width of 1 wave number.

Example: NOTEBOOK SEASONAL_CYCLE_FILTERING_FFT.IPYNB

Possible problems:

- cannot be done in real time
- a response function with a sharp cutoff introduces spurious behavior in the time domain (and vice versa), particularly at the end - so just doing a boxcar filter on the FFT coefficients is not typically a good idea unless the variance at higher frequencies is known to be very small

As mentioned earlier, bandpass filtering noise will make it appear periodic - so beware! Below is an example using filtering in the frequency domain to make this point.

Example: NOTEBOOK BANDPASSFILTERING_NOISE.IPYNB

1.6.3 Filtering in the time domain

In this case, you apply a weighting function (e.g. weighted running average) in the time domain, so that certain frequencies are removed. That is, you write

$$y(t) = \sum_{\tau=-J}^{J} g(\tau) \cdot x(t+\tau) \tag{65}$$

where

- $g(\tau)$ is the weighting function,

- $x(t)$ is the original time series,
- $y(t)$ is the filtered $x(t)$,
- τ denotes different lags
- the length of the weighting function g is $2 \cdot J + 1$, or, $J = (L-1)/2$

Note that J data points are lost at each end of the time series.

1-1-1 filter
As a reminder, a moving average of width $L = 3$ can be written in the following way,

$$y(t) = \frac{1}{3}x(t-\tau) + \frac{1}{3}x(t) + \frac{1}{3}x(t+\tau) \tag{66}$$

1-1-1-1-1 filter
A moving average of width $L = 5$ can be written in the following way,

$$y(t) = \frac{1}{5}x(t-2\tau) + \frac{1}{5}x(t-\tau) + \frac{1}{5}x(t) + \frac{1}{5}x(t+\tau) + \frac{1}{5}x(t+2\tau) \tag{67}$$

This method can only be used to *smooth* a time series, and thus, is always a low pass filter, with response function $R(\omega)_L$.

To find the high-pass filter component, you can just subtract the low-pass from the original:

$$R(\omega)_H = 1 - R(\omega)_L \tag{68}$$
$$y(t)_{H,\mathtt{filtered}} = x(t) - y(t)_{L,\mathtt{filtered}} \tag{69}$$

A bandpass filter can be made by subtracting one low-pass filtered time series from another with a cutoff at a higher frequency.

$$y(t)_{3-6\mathtt{dys}} = y(t)_{L,3\mathtt{dys}} - y(t)_{L,6\mathtt{dys}} \tag{70}$$

What does a particular weighting function $g(\tau)$ do to the frequencies in our time series?

$$y(t) = \sum_{\tau=-J}^{J} g(\tau) \cdot x(t+\tau) \tag{71}$$

this is just the convolution of x and g, written another way,

$$y(t) = \int_{-\infty}^{\infty} x(t+\tau)g(\tau)d\tau = x * g \tag{72}$$

Thus,

$$\mathcal{F}(y(t)) = \mathcal{F}(x * g) = \mathcal{F}(x) \cdot \mathcal{F}(g) = X \cdot G \tag{73}$$

That is, the Fourier transform of the smoothed time series is the product of the Fourier transform of the original (unsmoothed/unfiltered) time series multiplied by the Fourier transform of the weighting function g.

Taking things one step further,

$$Y_{filtered}(\omega) = X(\omega) \cdot G(\omega) \tag{74}$$

$$R(\omega) = \frac{\text{amplitude of filtered series at } \omega}{\text{amplitude of original series at } \omega} = \frac{Y_{filtered}(\omega)}{X(\omega)} = G(\omega) \tag{75}$$

that is, the response function of the smoothing is the Fourier transform of the weighting function g - they constitute a Fourier Transform pair.

Example weighting function: the running mean

$$g(\tau) = 1/L \text{ for } -\frac{L-1}{2} \leqslant \tau \leqslant \frac{L-1}{2} \tag{76}$$

$$g(\tau) = 0 \text{ for all other } \tau \tag{77}$$

where L is the length of the window.

The response function is the Fourier transform of $g(\tau)$, which is

$$R(\omega) = G(\omega) = \frac{\sin(\omega L/(2))}{\omega L/(2)} = \mathrm{sinc}(\omega L/(2\pi)) \tag{78}$$

We already saw this exact formulation in terms of windowing our time series, but now it appears again in the context of moving box averages!

Note that

$$R(\omega) = 0 \tag{79}$$

when $\omega \cdot L = 2\pi, 4\pi, 6\pi$, or when $\omega = 2\pi/L, 4\pi/L...$ - that is, when you have 1 cycle/L, 2 cycles/L, 3 cycles/L.

So, if you apply a 12-month running mean to monthly data, you will *exactly* remove the 12-month cycle, the 6 month cycle, the 4 month cycle, the 2 month cycle, ...

However, the problem is the side lobes of the sinc function! These will phase shift your frequencies between these perfectly removed harmonics, leading to spectral leakage to other frequencies.

Example weighting function: the perfect response function
We have already seen this one!

Perfect $R(\omega)$: this is a rectangle around the frequencies you wish to keep

Time domain: the window in the time domain is once again a sinc function, note that the window extends over a wide range of the time series, and negatively counts time steps in the negative lobes...this can cause huge issues with the data at the ends.

So, like with windowing, the best solution is something in the middle...i.e. a tapered window.

Example weighting function: the Gaussian Bell
A Gaussian,

$$f(x) = e^{-x^2} \tag{80}$$

forms a Fourier Transform pair with itself. So if, instead of a rectangular moving average in time, one can do a Gaussian moving average in time. This is used quite often when doing spatial averaging, although note that the response function doesn't fall off terribly quickly (not a very sharp cutoff).

1.6.4 Creating your own non-recursive filters

In general, you can create your own symmetric filter by:

1. generating a desired $R(\omega)$ and then
2. Fourier transforming $R(\omega)$ to get your time window $g(\tau)$

Applying filters in a row
If you find a filter/smoother that you like, you can also apply it multiple times in a row (repeated applications). For example, say you apply a filter $w_1(\tau)$ to your data, then you apply $w_2(\tau)$ to the filtered data from the first step, in this case (using our convolution theorem),

$$\begin{aligned} \mathcal{F}(y_1(t)) &= \mathcal{F}(y(t)) \cdot \mathcal{F}(w_1(\tau)) & (81)\\ \mathcal{F}(y_2(t)) &= \mathcal{F}(y_1(t)) \cdot \mathcal{F}(w_2(\tau)) & (82)\\ \mathcal{F}(y_2(t)) &= \mathcal{F}(y(t)) \cdot \mathcal{F}(w_1(\tau)) \cdot \mathcal{F}(w_2(\tau)) & (83) \end{aligned}$$

That is, the total response function is just the product of the separate response functions!

Response function of moving averages (both tapered and not tapered)
Often, one wants a simple method for smoothing their data, and they gravitate to the moving average. However, we have shown that the response function of the moving average is not ideal. However, there are nearly as simple types of moving averages with much more desirable response functions.

Next, we will introduce a symmetric, but *not* rectangular moving average. That is, the 1-2-1 filter:

1-2-1 filter

$$y(t) = \frac{1}{4}x(t-\tau) + \frac{1}{2}x(t) + \frac{1}{4}x(t+\tau) \tag{84}$$

How can we determine the response function of such a simple filter?

1. run it through an FFT and see what comes out
2. use the simple formula below...

Say we have a symmetric filter:

$$y(t) = \sum_{\tau=-J}^{J} C_\tau \cdot x(t-\tau) \tag{85}$$

$$C_{-\tau} = C_\tau \tag{86}$$

What is $Y(\omega)$ (the FT of y)?

$$\mathcal{F}(y(t)) = \mathcal{F}\left(\sum_{\tau=-J}^{J} C_\tau \cdot x(t-\tau)\right) \tag{87}$$

$$Y(\omega) = \int_{-\infty}^{\infty} \left(\sum_{\tau=-J}^{J} C_\tau \cdot x(t-\tau)\right) e^{-i\omega t} dt \tag{88}$$

We can take the sum out of the integral since it is over τ and not t.

$$Y(\omega) = \sum_{\tau=-J}^{J} \left(C_\tau \cdot \int_{-\infty}^{\infty} x(t-\tau) \cdot e^{-i\omega t} dt\right) \tag{89}$$

The integrand almost just looks like the Fourier Transform of x, but not quite because of that pesky τ.

This is when the Time Shifting Theorem comes to the rescue.

Time Shifting Theorem

$$\mathcal{F}(x(t-a)) = \mathcal{F}(x(t)) \cdot e^{-i\omega a} \tag{90}$$

(note this is actually quite straight-forward to prove, so I won't do it here).

Thus,

$$Y(\omega) = \left(\sum_{\tau=-J}^{J} C_\tau e^{-i\omega\tau}\right) \cdot X(\omega) \tag{91}$$

Thus, the response function is given by

$$R(\omega) = \frac{Y(\omega)}{X(\omega)} = \sum_{\tau=-J}^{J} C_\tau e^{-i\omega\tau} \tag{92}$$

Using the fact that the C_τs are symmetric, and the fact that $\cos(x) = \frac{e^{ix}+e^{-ix}}{2}$ we can rewrite this as

$$R(\omega) = \frac{Y(\omega)}{X(\omega)} = C_0 + 2 \cdot \sum_{\tau=1}^{J} C_\tau \cdot \cos(\omega\tau) \tag{93}$$

What does this mean for us? Let's do some examples:

Worked Example: Example 4.3

Example: Figure for Example 4.3

Example: notebook response_functions_moving_agv.ipynb

Example 4.3, page 1 of 2

Response functions of simple filter weights

$$y(t) = \sum_{\tau=-J}^{J} C_\tau \cdot x(t-\tau) \tag{94}$$

$$C_{-\tau} = C_\tau \tag{95}$$

$$R(\omega) = C_0 + 2 \cdot \sum_{\tau=1}^{J} C_\tau \cdot \cos(\omega\tau) \tag{96}$$

Using the above equations for a non-recursive filter and its response function, find the response functions of the following filters:

Running mean 3: 1-1-1 filter

$$C_{\tau=-1} = 1/3, C_{\tau=0} = 1/3, C_{\tau=+1} = 1/3 \tag{97}$$

$$R(\omega) = \frac{1}{3} + 2 \cdot \sum_{\tau=1}^{1} C_\tau \cdot \cos(\omega\tau) \tag{98}$$

$$= \frac{1}{3} + 2 \cdot \frac{1}{3} \cdot \cos(\omega) \tag{99}$$

$$= \frac{1}{3} + \frac{2}{3} \cdot \cos(\omega) \tag{100}$$

Running mean 5: 1-1-1-1-1 filter

$$C_{\tau=-2} = 1/5, C_{\tau=-1} = 1/5, C_{\tau=0} = 1/5, C_{\tau=+1} = 1/5, C_{\tau=+2} = 1/5 \tag{101}$$

$$R(\omega) = \frac{1}{3} + 2 \cdot \sum_{\tau=1}^{2} C_\tau \cdot \cos(\omega\tau) \tag{102}$$

$$= \frac{1}{5} + 2 \cdot \frac{1}{5} \cdot \cos(\omega) + 2 \cdot \frac{1}{5} \cdot \cos(2\omega) \tag{103}$$

$$= \frac{1}{3} + \frac{2}{5} \cdot \cos(\omega) + \frac{2}{5} \cdot \cos(2\omega) \tag{104}$$

1-2-1 filter

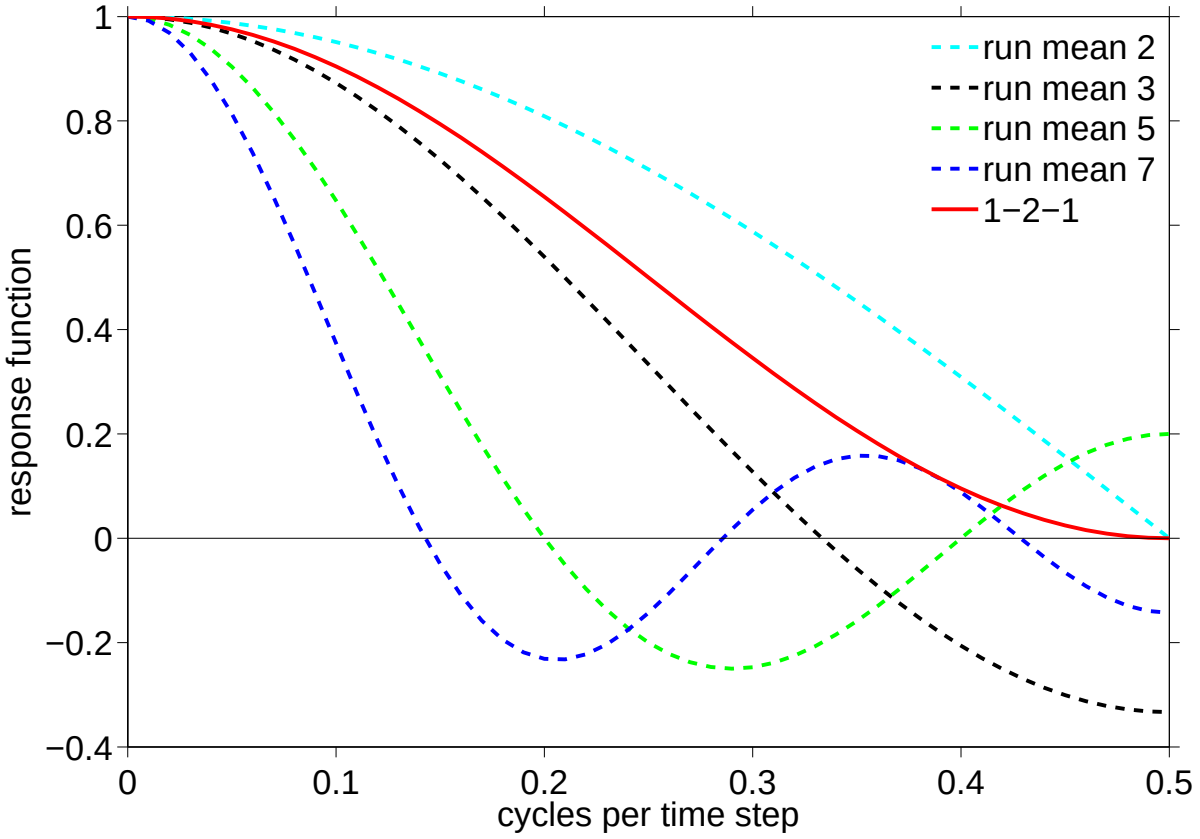
$$C_{\tau=-1} = 1/4, C_{\tau=0} = 1/2, C_{\tau=+1} = 1/4 \tag{105}$$

$$R(\omega) = \frac{1}{2} + 2 \cdot \sum_{\tau=1}^{1} C_\tau \cdot \cos(\omega\tau) \tag{106}$$

$$= \frac{1}{2} + 2 \cdot \frac{1}{4} \cdot \cos(\omega) \tag{107}$$

$$= \frac{1}{2} + \frac{1}{2} \cdot \cos(\omega) \tag{108}$$

Example 4.3, page 2 of 2



Lanczos smoothing
One type of filter is known as a low-pass Lanczos filter:
Lanczos filter: derived by designing a filter with as sharp of a cutoff as possible, but with some tweaks to the filter weights to remove the harmonic lobes from the sinc function (Gibb's phenomenon)

The idea is that one can write down an equation using J filter weights that gives the most abrupt frequency cutoff, specifically,

$$R(\omega) = C_0 + 2 \cdot \sum_{\tau=1}^{J} C_\tau \cdot \cos(\omega\tau) \tag{109}$$

$$C_\tau = \frac{1}{\pi} \int_0^\pi \cos(\tau\omega') R(\omega') d\omega' \tag{110}$$

where $R(\omega)$ is the response function that you want to get!

For example, if you want an abrupt cutoff at $\omega = \alpha\pi$.

$$C_\tau = \frac{1}{\tau\pi} \sin(\alpha\tau\pi) \tag{111}$$

The sharper you want the cutoff, the more weights you need. However, in all cases, you will get wiggles from the sharp edges of the filter (Gibbs phenomena). Thus, special weights, called "Lanczos weights" are implemented to allow a small number of weights to be used (small J) but to remove the Gibbs phenomena. (Note that fewer weights means faster computation time).

The Lanczos weights do just this, and are sinc functions. Thus, the new filter weights, $\tilde{C}_\tau$ are:

$$\tilde{C}_\tau = \mathrm{sinc}\left(\frac{\pi\tau}{J}\right) C_\tau \tag{112}$$

Example: SHOW FIGURES FROM HARTMANN NOTES

1.6.5 Recursive filters

Recursive filters have a few basic properties that often make them ideal filters:

- they allow for less computation time, as you use the already filtered data rather than the raw data at each step
- similar to above, recursive filters require fewer weights to get the job done
- can be done in real time using only the previous data

However, recursive filter construction is not always easy, and does not always lead to stable results!

For a non-recursive filter:

$$y(t) = \sum_{\tau=-J}^{J} g(\tau) \cdot x(t+\tau) \tag{113}$$

For a recursive filter:

$$y(t) = \sum_{\tau=-J}^{0} b_\tau \cdot x(t+\tau) + \sum_{\tau=-J}^{-1} a_\tau \cdot y(t+\tau) \tag{114}$$

We won't go into detail here, but a very important aspect of a recursive filter is how long it takes the influence of a certain data point to die away - this can be calculated with the "impulse response". This calculation is critical, since the beginning of your time series (when you are spinning up the recursive filter) can corrupt your data. So, you always want to remove the first M data points, where M is the number of days it takes the impulse response function to become negligible.

Butterworth filters
A well-known (and used) family of recursive filters are the Butterworth filters, designed to

- be smooth
- have high tangency at the origin and infinity
- have as flat a frequency in the passband as possible

The response function of the Butterworth filter is written as:

$$|R(\omega)|^2 = R(\omega)R^*(\omega) = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}} \tag{115}$$

where $N+1$ is the number of weights (a's and b's) and w_c is the cutoff frequency. Note that as N goes to infinity, the response function approaches a boxcar function.

In Matlab, *butter.m* outputs the weights a and b that can be applied using *filter.m*.

Example: NOTEBOOK BUTTERWORTH_LANCZOS_EXAMPLE.IPYNB

1.7 Cross-spectrum Analysis

Cross-spectrum analysis is used when one has two time series and wants to determine their relationship as a function of frequency. Sometimes, this comes about because both series have spectral peaks around a similar frequency, and one wants to assess the phase relationship between the two.

On the other hand, imagine that neither series has any spectral peaks - but, lurking in the background could still be coherent modes. Cross-spectrum analysis can draw these out for you.

In essence, the goal is the following: *Given two time series* $x(t)$ *and* $y(t)$*, find the frequency and phase of their relationship.*

For example, imagine a propagating wave (e.g. the MJO), the winds at one location and one further east may oscillate at the same frequency, but may be out of phase by 90 degrees.

Example: SHOW SLIDES 17_INTRO_CROSS_SPECTRAL_ANALYSIS.PDF

Example: NOTEBOOK CROSS_SPECTRUM_EXAMPLE.IPYNB

1.7.1 Cross-spectrum, co-spectrum and quadrature spectrum

While there are good canned software packages out there for cross-spectrum analysis, it is still important to understand where they come from, and what types of choices you have to make to use them. Thus, we will go over the basics, without going too far into the gory details.

To simplify (and shorten) the notation, we will define the argument of the cosines and sines as:

$$\alpha k = 2\pi k \frac{t}{T} \tag{116}$$

From the discrete Fourier transform, a pair of time series x and y can be decomposed such that:

$$x(t) = \overline{x} + \left(\sum_{k=1}^{N/2-1} A_{xk} \cos(\alpha k) + \sum_{k=1}^{N/2-1} B_{xk} \sin(\alpha k) \right) + A_{xN/2} \cos(\alpha N/2) \tag{117}$$

$$y(t) = \overline{y} + \left(\sum_{k=1}^{N/2-1} A_{yk} \cos(\alpha k) + \sum_{k=1}^{N/2-1} B_{yk} \sin(\alpha k) \right) + A_{yN/2} \cos(\alpha N/2) \tag{118}$$

Using the fact that the cosine and sine functions are orthogonal to each other, we can write the covariance of x and y as:

$$\overline{x'y'} = \overline{\sum_{k=1}^{N/2-1} A_{xk}A_{yk} \cos^2(\alpha k) + \sum_{k=1}^{N/2-1} B_{xk}B_{yk} \sin^2(\alpha k) ...} \tag{119}$$

$$... + \overline{A_{xN/2}A_{yN/2} \cos^2(\alpha N/2)} \tag{120}$$

Recall that the variance of the sines and cosines (the squared terms under the average) are $1/2$ for $k \neq N/2$, and the A's and B's are not a function of time, so,

$$\overline{x'y'} = \frac{1}{2}\sum_{k=1}^{N/2-1}(A_{xk}A_{yk} + B_{xk}B_{yk}) + A_{xN/2}A_{yN/2} \quad (121)$$

$$= \sum_{k=1}^{N/2} CO(k) = \text{cospectrum of x and y} \quad (122)$$

$CO(k)$ is the co-spectrum of x and y as a function of k.

Note that in the special case where $x = y$,

$$\overline{x'^2} = \frac{1}{2}\sum_{k=1}^{N/2-1}\left(A_k^2 + B_k^2\right) + A_{N/2}^2 = \sum_{k=1}^{N/2}\frac{C_k^2}{2} \quad (123)$$

Next, we define the following complex quantities (you'll see why in a minute):

$$F_x(k) = (A_{xk} - iB_{xk}) \cdot e^{i(\alpha k)} = C_{xk} \cdot e^{-i\theta_{xk}}e^{i(\alpha k)} \quad (124)$$

$$F_y(k) = (A_{yk} - iB_{yk}) \cdot e^{i(\alpha k)} = C_{yk} \cdot e^{-i\theta_{yk}}e^{i(\alpha k)} \quad (125)$$

Note that $F_x(k)$ and $F_y(k)$ can be decomposed into real and imaginary parts using the fact that $e^{i\theta} = \cos(\theta) + i\sin(\theta)$. For example,

$$F_x(k) = (A_{xk} - iB_{xk})(\cos(\alpha k) + i\sin(\alpha k)) \quad (126)$$

$$= A_{xk}\cos(\alpha k) + B_{xk}\sin(\alpha k)...[\textbf{Re part}] \quad (127)$$

$$+ \; i(A_{xk}\sin(\alpha k) - B_{xk}\cos(\alpha k))\,[\textbf{Im part}] \quad (128)$$

From the above, it follows that x and y can be written as

$$x = \overline{x} + \sum_{k=1}^{k=N/2}\textbf{Re}(F_x(k)) \quad (129)$$

$$y = \overline{y} + \sum_{k=1}^{k=N/2}\textbf{Re}(F_y(k)) \quad (130)$$

Next, define the following three quantities:

$$F_{xx} = F_x(k)F_x(k)^* \quad (131)$$

$$F_{yy} = F_y(k)F_y(k)^* \quad (132)$$

$$F_{xy} = F_x(k)F_y(k)^* \quad (133)$$

Remember that the complex conjugate of Z (that is, Z^*) is defined as

$$Z^* = (\mathbf{Re}(Z) - i\mathbf{Im}(Z)) \tag{134}$$

and thus, ZZ^* can be simplified to

$$\begin{aligned} ZZ^* &= (\mathbf{Re}(Z) + i\mathbf{Im}(Z)) \cdot (\mathbf{Re}(Z) - i\mathbf{Im}(Z)) && (135)\\ &= (\mathbf{Re}(Z))^2 - i\mathbf{Re}(Z)\mathbf{Im}(Z) + i\mathbf{Im}(Z)\mathbf{Re}(Z) + (\mathbf{Im}(Z))^2 && (136)\\ &= (\mathbf{Re}(Z))^2 + (\mathbf{Im}(Z))^2 && (137) \end{aligned}$$

$F_{xx}(k)$ (the math is similar for $F_{yy}(k)$) is then:

$$\begin{aligned} F_{xx}(k) &= F_x(k)F_x(k)^* = (\mathbf{Re}(F_x(k)))^2 + (\mathbf{Im}(F_x(k)))^2 && (138)\\ &= A_{xk}^2 \cos^2(\alpha k) + B_{xk}^2 \sin^2(\alpha k) + A_{xk}^2 \sin^2(\alpha k) + B_{xk}^2 \cos^2(\alpha k) && (139)\\ &= A_{xk}^2(\cos^2(\alpha k) + \sin^2(\alpha k)) + B_{xk}^2(\sin^2(\alpha k) + \cos^2(\alpha k)) && (140) \end{aligned}$$

thus,

$$\begin{aligned} F_{xx}(k) = A_{xk}^2 + B_{xk}^2 = C_{xk}^2 && (141)\\ F_{yy}(k) = A_{yk}^2 + B_{yk}^2 = C_{yk}^2 && (142) \end{aligned}$$

Note that this is just 2 times the spectrum of $x(t)$ and $y(t)$. That is,

$$\overline{x'^2} = \sum_{k=1}^{N/2} \frac{C_k^2}{2} = \sum_{k=1}^{N/2} \frac{F_{xx}(k)}{2} \tag{143}$$

Moving now to F_{xy},

$$\begin{aligned} F_{xy}(k) &= F_x F_y^* = [\mathbf{Re}(F_x) + i\mathbf{Im}(F_x)]\,[\mathbf{Re}(F_y) - i\mathbf{Im}(F_y)] && (144)\\ &= \mathbf{Re}(F_x)\mathbf{Re}(F_y) - i\mathbf{Re}(F_x)\mathbf{Im}(F_y) + i\mathbf{Im}(F_x)\mathbf{Re}(F_y) + \mathbf{Im}(F_x)\mathbf{Im}(F_y) && (145)\\ &= [\mathbf{Re}(F_x)\mathbf{Re}(F_y) + \mathbf{Im}(F_x)\mathbf{Im}(F_y)] + i\,[\mathbf{Im}(F_x)\mathbf{Re}(F_y) - \mathbf{Re}(F_x)\mathbf{Im}(F_y)] && (146) \end{aligned}$$

The real part of F_{xy} is given by,

$$\begin{aligned} \mathbf{Re}(F_{xy}) &= \mathbf{Re}(F_x) \cdot \mathbf{Re}(F_y) + \mathbf{Im}(F_x) \cdot \mathbf{Im}(F_y) && (147)\\ &= (A_{xk}\cos(\alpha k) + B_{xk}\sin(\alpha k))(A_{yk}\cos(\alpha k) + B_{yk}\sin(\alpha k))... && (148)\\ &+ (A_{xk}\sin(\alpha k) - B_{xk}\cos(\alpha k))(A_{yk}\sin(\alpha k) - B_{yk}\cos(\alpha k)) && (149)\\ &= A_{xk}A_{yk}\cos^2(\alpha k) + A_{xk}B_{yk}\cos(\alpha k)\sin(\alpha k) + B_{xk}A_{yk}\cos(\alpha k)\sin(\alpha k) + B_{xk}B_{yk}\sin^2(\alpha k)... && (150)\\ &+ A_{xk}A_{yk}\sin^2(\alpha k) - A_{xk}B_{yk}\cos(\alpha k)\sin(\alpha k) - B_{xk}A_{yk}\cos(\alpha k)\sin(\alpha k) + B_{xk}B_{yk}\cos^2(\alpha k) && (151)\\ &= A_{xk}A_{yk}(\cos^2(\alpha k) + \sin^2 \alpha k) + B_{xk}B_{yk}(\sin^2(\alpha k) + \cos^2 \alpha k) && (152)\\ &= A_{xk}A_{yk} + B_{xk}B_{yk} && (153) \end{aligned}$$

where to simplify notation we denote $F_x(k)$ as F_{xk}.

Following similar steps, the imaginary part of F_{xy} is given by

$$\mathbf{Im}(F_{xy}) = \mathbf{Re}(F_y) \cdot \mathbf{Im}(F_x) - \mathbf{Re}(F_x) \cdot \mathbf{Im}(F_y) = A_{xk}B_{yk} - A_{yk}B_{xk} \tag{154}$$

Therefore,

$$F_{xy}(k) = (A_{xk}A_{yk} + B_{xk}B_{yk}) + i(A_{xk}B_{yk} - A_{yk}B_{xk}) \tag{155}$$

From above,

$$\overline{x'y'} = \frac{1}{2}\sum_{k=1}^{N/2} \mathbf{Re}(F_{xy}(k)) \tag{156}$$

In words:

- $F_{xy}(k)$ is the cross-spectrum of x and y
- the **Re()** part of $F_{xy}(k)$ is the cospectrum - this is the in phase signal: $CO(k) = A_{xk}A_{yk} + B_{xk}B_{yk}$
- the **Im()** part of $F_{xy}(k)$ is the quadrature spectrum - this is the out of phase signal: $Q(k) = A_{xk}B_{yk} - A_{yk}B_{xk}$
- the amplitude of the cross-spectrum is given by $|F_{xy}| = C_{xk}C_{yk}$ and the phase by $\theta_{xyk} = \theta_{yk} - \theta_{xk}$ (think Pythagorean Theorem)

1.7.2 Coherence squared

The coherence is defined such that:

$$Coh^2(k) = \frac{|F_{xy}|^2}{F_{xx}F_{yy}} \tag{157}$$

IMPORTANT: The Coh^2 is equal to 1 at any specific value of $k = k_1$. (see below)

$$\begin{aligned}
Coh^2(k = k_1) &= \frac{|F_{xy}|^2}{F_{xx}F_{yy}} = \frac{F_{xy}F^*_{xy}}{F_{xx}F_{yy}} = \frac{CO(k)^2 + Q(k)^2}{C_x^2 C_y^2} && (158)\\
&= \frac{(A_xA_y + B_xB_y)^2 + (A_xB_y - A_yB_x)^2}{(A_x^2 + B_x^2)(A_y^2 + B_y^2)} && (159)\\
&= \frac{A_x^2A_y^2 + 2A_xA_yB_xB_y + B_x^2B_y^2 + A_x^2B_y^2 - 2A_xB_yA_yB_x + A_y^2B_x^2}{A_x^2A_y^2 + A_x^2B_y^2 + B_x^2A_y^2 + B_x^2B_y^2} && (160)\\
&= \frac{A_x^2A_y^2 + A_x^2B_y^2 + B_x^2A_y^2 + B_x^2B_y^2}{A_x^2A_y^2 + A_x^2B_y^2 + B_x^2A_y^2 + B_x^2B_y^2} && (161)\\
&= 1 && (162)
\end{aligned}$$

where we have used the fact that $(a + bi) \cdot (a - bi) = a^2 + b^2$ to simplify the numerator.

However, when Coh^2 is calculated from an average cross-spectrum, say, from averaging over multiple chunks or averaging over successive wave numbers, it is no longer constrained to be 1. This is shown in detail in Hartmann's notes, however, we can go through a bit of it here.

Suppose we have two chunks of x and y each (denoted as $m = 1$ and $m = 2$) from which we are calculating the average cross-spectrum ($\overline{F_{xy}}$). Note this exercise also works for averaging two adjacent spectral coefficients together in the same chunk too. It can be shown that the Coh^2 calculated from this average cross-spectrum is equal to

$$Coh^2 = \frac{C_{x1}^2 C_{y1}^2 + 2C_{x1}C_{y1}C_{x2}C_{y2}\cos(\theta_1 - \theta_2) + C_{x2}^2 C_{y2}^2}{C_{x1}^2 C_{y1}^2 + C_{x1}^2 C_{y2}^2 + C_{x2}^2 C_{y1}^2 + C_{x2}^2 C_{y2}^2} \tag{163}$$

Things to notice:

- In practice, for 2 unrelated time series, Coh^2 drops off rapidly as it is averaged over multiple chunks (or a range of k).
- the coherence is largest when the phase difference is smallest, that is, $\theta_1 = \theta_2$. So, if the phase relationships between x and y at frequency k is different between the two chunks then the average Coh^2 will be small. This physically makes sense since we expect similar phase relationships between chunks if the relationship is real/physical.
- Suppose the phase difference is the same, we still do not have perfect coherence. The coherence will be maximized if $\frac{C_{x1}}{C_{y1}} = \frac{C_{x2}}{C_{y2}}$, that is, if the amplitude ratios between x and y are the same for the two chunks. This says that x and y are linearly related to each other.

> The coherence will be 1 when realizations (or spectral estimates) are averaged, only under the conditions that the two realizations averaged together show the same phase difference between the two variables and the same amplitude ratio between the two variables. The coherence thus shows how well these two conditions are satisfied.

Here's, how it is used in practice.

Assume you have divided your data into M chunks.

1. calculate the FFT of x and y to get the A's and B's
2. calculate the $CO(k)$, $Q(k)$, $C_x(k)^2$ and $C_y(k)^2$ *for each subset of the data*
3. calculate Coh^2 the following way

$$\mathsf{Coh}^2(\mathsf{k}) = \frac{\left(\frac{1}{\mathsf{M}}\sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{CO}(\mathsf{k})\right)^2 + \left(\frac{1}{\mathsf{M}}\sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{Q}(\mathsf{k})\right)^2}{\left(\frac{1}{\mathsf{M}}\sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{C}_{\mathsf{x}}^2(\mathsf{k})\right)\left(\frac{1}{\mathsf{M}}\sum_{\mathsf{i}=1}^{\mathsf{M}} \mathsf{C}_{\mathsf{y}}^2(\mathsf{k})\right)} \tag{164}$$

When averaged over chunks (or a range of k), the $\mathsf{Coh}^2(\mathsf{k})$ gives the fraction of the variance of $\mathsf{y}(\mathsf{t})$ that can be explained using $\mathsf{x}(\mathsf{t})$ as a predictor over that frequency band. Note that the Coh^2 looks very similar to R^2 from linear theory:

$$\mathsf{R}^2 = \frac{\left(\overline{\mathsf{x}'\mathsf{y}'}\right)^2}{\overline{\mathsf{x}'^2} \cdot \overline{\mathsf{y}'^2}} \tag{165}$$

The significance of the Coh^2 is a function of the degrees of freedom available for the individual spectral estimates. Most tables are given in terms of M, the number of k averaged over in the individual spectral estimates (or M, the number of chunks). So, although the degrees of freedom is roughly $2 \cdot \mathsf{M}$ as usual, you should only use M in these types of lookup tables!

1.7.3 Getting the phase information

The phase of the average cross-spectrum reveals the phase lag (if any) between x and y for a given k, but is only meaningful if the Coh^2 is meaningful. You can obtain the phase difference between x and y for a given wavenumber k by using the Pythagorean Theorem:

$$\tan(\theta_{\mathsf{xk}} - \theta_{\mathsf{yk}}) = \frac{\overline{\mathsf{Q}(\mathsf{k})}}{\overline{\mathsf{CO}(\mathsf{k})}} = \frac{\overline{\mathbf{Im}(\mathsf{F}_{\mathsf{xy}})}}{\overline{\mathbf{Re}(\mathsf{F}_{\mathsf{xy}})}} \tag{166}$$

Example: RE-RUN NOTEBOOK CROSS_SPECTRUM_EXAMPLE.IPYNB

1.8 Mixed Space-Time Analysis

You can think of spectral analysis in terms of the following family of methods:

- spectral analysis (one variable; FFT one dimension)
- cross-spectral analysis (two variables; FFT one dimension)
- mixed space/time spectral analysis (one variable; FFT two dimensions)
- mixed space/time cross-spectral analysis (two variables; FFT two dimensions)

The details of the algebra involved with space-time spectral analysis can be found in a series of papers by Hayashi, Y.:

- Journal of the Meteorological Society of Japan (1971)
- Journal of the Atmospheric Sciences (1977)
- Journal of the Meteorological Society of Japan (1982)

1.8.1 Space-Time Spectral Analysis

The basic idea of mixed space/time analysis:

In the case of spectral analysis, we decompose $y(t)$ into contributions from different temporal harmonics:

$$y(t) = \overline{y} + \sum_{k=1}^{N/2} A_k \cos(2\pi k t/T) + \sum_{k=1}^{N/2} B_k \sin(2\pi k t/T) \tag{167}$$

(where I have omitted the special case of the Nyquist frequency to make the writing easier).

This can be written in phase/amplitude form as:

$$y(t) = \overline{y} + \sum_{k=1}^{N/2} C_k \cos(2\pi k t/T - \phi_k) \tag{168}$$

The summation can be easily converted to frequency (rather than wavenumber) as:

$$y(t) = \overline{y} + \sum_{\omega=\frac{2\pi}{T}}^{\pi} W_\omega \cos(\omega t + \phi_\omega) \tag{169}$$

Where T is the length of the data record and W is the weights associated with each harmonic.

In the case of mixed space/time spectral analysis, we start with data $y(x,t)$, that is, y is made of both a time and a space dimension. In this case, we decompose y into contributions from different temporal and spatial harmonics.

It is most common to use space/time spectral analysis in the case where the time harmonics are frequency and the spatial harmonics correspond to waves along latitude circles. (Note, if you use latitude circles, you don't need to window in this dimension since you data is actually periodic!)

$$y(t,\lambda) = \overline{y} + \sum_{k=1}^{n/2} \sum_{\omega=\pm\frac{2\pi}{T}}^{\pi} W_{k,\pm\omega} \cos\left(\pm\omega t + \lambda k + \phi_{k,\pm\omega}\right) \tag{170}$$

where

- T is the length of the record ($0 \leqslant t \leqslant T$)
- λ is the longitude in radians ($0 \leqslant \lambda \leqslant 2\pi$)
- k is the zonal wave number (number of times the wave fits around a latitude circle)
- n is the number of grid points along a latitude circle

ω can be positive and negative at each value of k since the phase speed of a wave can be positive (westward moving) or negative (eastward moving). Recall that

$$c_{ph} = \omega/k \tag{171}$$

The goal of space-time spectral analysis is to find the weights $W(\omega, k)$.

In words, here are the steps:

1. Decompose data y into zonal wave numbers $k = 1$ to $n/2$. You will have sin and cos functions at each time step t.

   Let $\mathbf{Y}(\lambda, t) = (144{,}3650)$.
   After decomposing into zonal wave numbers, you will have two matrices, a matrix for the cosine functions and one for the sine functions.
   $\mathbf{X}_C(k,t) = (72, 3650)$ and $\mathbf{X}_S(k,t) = (72, 3650)$

2. Fourier transform $\mathbf{X}_C$ and $\mathbf{X}_S$ in time.

   In our example, you will have:

$$\mathbf{X}_C(k,t) = \sum_{\omega=\frac{2\pi}{T}}^{\pi} A_{k,\omega} \cos(\omega t) + B_{k,\omega} \sin(\omega t) \tag{172}$$

$$\mathbf{X}_S(k,t) = \sum_{\omega=\frac{2\pi}{T}}^{\pi} a_{k,\omega} \cos(\omega t) + b_{k,\omega} \sin(\omega t) \tag{173}$$

3. The regression coefficients $A_{k,\omega}$, $B_{k,\omega}$, $a_{k,\omega}$ and $b_{k,\omega}$ are related to $W_{k,\pm\omega}$ and $\phi_{k,\pm\omega}$ through the following relationship:

$$4W^2_{k,\pm\omega} = (A \mp b)^2 + (\mp B - a)^2 \tag{174}$$

$$\phi_{k,\pm\omega} = a \tan\left(\frac{\mp B - a}{A \mp b}\right) \tag{175}$$

and the power spectrum is equal to:

$$P_{k,\pm\omega} = \frac{1}{2}W^2_{k,\pm\omega} \tag{176}$$

4. It is common to plot the resulting power spectrum as a function of frequency on the x-axis and spatial wave number on the y-axis.

1.8.2 Space-Time Cross-Spectral Analysis

The methodology is the same as for mixed space-time spectral analysis, but now you are investigating relationships between two variables, say u' and v'.

To calculate the mixed space/time cross-spectra between two data sets $X(x,t)$ and $Y(x,t)$:

1. Calculate $W_{k,\pm\omega}$ and $\phi_{k,\pm\omega}$ for X and Y separately.

   Call the amplitudes and phases: $W^X_{k,\pm\omega}, W^Y_{k,\pm\omega}, \phi^X_{k,\pm\omega}, \phi^Y_{k,\pm\omega}$.

2. For each k and ω pair, the co-spectrum and quadrature spectrum are found as:

$$Co_{k,\pm\omega} = \frac{1}{2}W^X_{k,\pm\omega}W^Y_{k,\pm\omega}\cos\left(\phi^X_{k,\pm\omega} - \phi^Y_{k,\pm\omega}\right) \tag{177}$$

$$Q_{k,\pm\omega} = \frac{1}{2}W^X_{k,\pm\omega}W^Y_{k,\pm\omega}\sin\left(\phi^X_{k,\pm\omega} - \phi^Y_{k,\pm\omega}\right) \tag{178}$$

$$\tag{179}$$

3. The average quadrature and co-spectra over a series of spatial, or more commonly temporal harmonics. This can be done over subsets of the data, or adjacent frequencies.

4. Finally, for each spatial and temporal harmonic, the coherence squares is given as

$$Coh^2(k, \pm\omega) = \frac{Co^2_{k,\pm\omega} + Q^2_{k,\pm\omega}}{P^X_{k,\pm\omega}P^Y_{k,\pm\omega}} \tag{180}$$

As an example, say you want to make winter-time plots of $u'v'$ given as a function of phase speed (related to wave number) and latitude, as done in Randel & Held, 1991.

1. For each winter (say, DJFM) you have 120 days. These are your chunks.
2. Let's focus on a single latitude for now. So, say, you're using NCEP data:
   $u = u(144, 120), v = v(144, 120)$.
3. First, FFT the data in space to get four matrices (two for u and two for v). The new matrices will have dimensions (72,120) since 72 wave numbers are resolved.
4. Then, FFT the sine and cosine matrices in time. So now you have four matrixes for u and four for v. Example: you will have a temporal sine and temporal cosine matrix for each spatial sine matrix, etc.
5. From the temporal cosine and sine matrices, you solve for W and ϕ for u and v separately.
6. Then you solve for CO and Q as given above.
7. Now you have CO and Q at a single latitude. Repeat the procedure for all latitudes. You will have $CO(\mathtt{latitude}, k, \omega)$ and $Q(\mathtt{latitude}, k, \omega)$.
8. Repeat the procedure for all of your winters (all of your chunks), and average the CO's and Q's together to get a mean CO and mean Q.
9. Now calculate Coh^2 from the averaged CO and Q.
10. You can choose to plot the Coh^2, or you can plot the *in-phase, cospectra power density:*

$$CPSD_{k,\pm\omega} = 2\overline{\mathbf{Re}(F_{X,k,\pm\omega}F^*_{Y,k,\pm\omega})} = 2 \cdot CO_{lat,\pm\omega} \tag{181}$$

    where the over line denotes the averages over many chunks or multiple bandwidths.

    Note that the CPSD can be negative, implying, for a example, a negative zonal momentum flux.

11. Now you have Coh^2 or the CPSD as a function of (latitude, k and ω) for all of your data. You can convert this to (latitude, phase speed) using the fact that $c_{ph} = \omega/k$.