Chapter 8
Filtering

8.1 Introduction

In this chapter we will consider the problem of filtering time or space series so that certain frequencies or wavenumbers are removed and some are retained. Filtering is an often used and sometimes abused method of accentuating certain frequencies and removing others. The technique can be used to isolate frequencies that are of physical interest from those that are not. It can be used to remove high frequency noise or low frequency trends from time series and leave unaltered the frequencies of interest. These applications are called low-pass and high-pass filtering, respectively. A band-pass filter will remove both high frequencies and low frequencies and leave only frequencies in a band in the middle. Band-pass filters tend to make even noise look periodic, or at least quasi-periodic, so one should verify that the frequency range of interest has some physically meaningful content before selecting it with a band-pass filter. We will begin by noting a few important theorems that constitute the fundamental tools of filtering.

8.1.1 The Convolution Theorem

If two functions $f_1(t)$ and $f_2(t)$ have Fourier Transforms $F_1(\omega)$ and $F_2(\omega)$, then the Fourier transform of the product of $f_1(t)$ and $f_2(t)$ is

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} F_1(\lambda)\, F_2(\omega-\lambda)\, d\lambda \tag{8.1}$$

and the Fourier transform of

$$\int_{-\infty}^{\infty} f_1(\tau)\, f_2(t-\tau)\, d\tau \tag{8.2}$$

is $F_1(\omega) \times F_2(\omega)$. This latter result is the most useful in filtering, since it says that the Fourier transform of the convolution of two functions in time is just the product of the Fourier transforms of the two individual functions.

8.1.2 Parseval's Theorem

Parseval's theorem states that

$$\int_{-\infty}^{\infty} f_1(t)\, f_2(t)\, dt = \frac{1}{2\pi}\int_{-\infty}^{\infty} F_2(\omega)\, F_1(\omega)^*\, d\omega \tag{8.3}$$

Here $F_1(\omega)^*$ indicates the complex conjugate of $F_1(\omega)$. For the case where $f_1(t) = f_2(t) = f(t)$, Parseval's theorem yields,

$$\int_{-\infty}^{\infty} f(t)^2 \, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(\omega)|^2 \, d\omega \tag{8.4}$$

Equation 8.4 equates the integral over time of the variance to the integral over frequency of the power. Thus the area under the power spectrum plotted versus frequency is equal to the variance of the time series.

8.2 Filtering

Suppose we wish to modify oscillations of certain frequencies in a time series while keeping other frequencies the same. For example, remove high frequency oscillations (low-pass filter), remove low frequencies (high-pass filter), or both (band-pass filter). The ozone layer of Earth s atmosphere is a low-pass filter for sunlight in the sense that it absorbs all energy with wavelengths shorter than 300 nanometers before it reaches the surface. Several different approaches to filtering can be taken.

8.2.1 Fourier Method

One possible method is to Fourier transform the timeseries of interest, multiply the Fourier coefficients by a suitable set of weights to remove or amplify the frequencies of interest, then reconstitute the time series by inverting the Fourier transform of the modified Fourier coefficients to produce the filtered timeseries. Here we show the proposed mathematical operation for a timeseries represented by a cosine series.

$$f(t) = \sum_{i=0}^{M} C_{\omega_i} \cos(\omega_i t - \phi_i) \tag{8.5}$$

$$f_{filtered}(t) = \sum_{i=0}^{M} R(\omega_i) \times C_{\omega_i} \cos(\omega_i t - \phi_i) \tag{8.6}$$

Here $R(\omega)$ is the amplitude response function for our filter, which would typically vary from zero (removal of the frequency ω) to one (passing the frequency through the filter unchanged).

$$R(\omega) = \frac{C_{\omega-filtered}}{C_{\omega-original}} \tag{8.7}$$

The problem with this method is that the reconstructed time series may not resemble the original one, particularly near the ends. This is the same general characteristic of functional fits discussed in an earlier chapter. Also, you need the whole record of data before you can produce a single filtered data point, and the most recently acquired values are at the end of the data stream, where the problems with the Fourier method are worst.

For realistic applications we most often use a local system of weights, so that the filtered timeseries always resembles the original timeseries at each point. These filter weight methods can be recursive, in which the already filtered data points are used in the filter, or non-recursive, in which only original non-filtered data are used to construct the filtered time series.

8.2.2 Centered, Non-recursive Filtering Method

A good place to start is with centered, non-recursive weights, which will introduce us to filtering with a simple but practical method. In this method, the original time series is subjected to a weighted running average, so that the filtered point is a weighted sum of surrounding points.

$$f_{filtered}(t) = \sum_{k=-J}^{J} w_k \, f(t + k\Delta t) \tag{8.8}$$

Some data points will be lost from each end of the time series since we do not have the values to compute the smoothed series for $i < J$ and $i > N - J$. It seems obvious that such an operation can produce only smoothed time series if the weights are positive and hence constitutes a low-pass filter. However, a high-pass filter can be constructed quite simply by subtracting the low-pass filtered time series from the original time series. The new high-pass response function will then be,

$$R_H(\omega) = 1 - R_L(\omega) \tag{8.9}$$

Where the subscripts H and L refer to high- and low-pass filters. One can then design a high-pass filter by first designing a low-pass filter that removes just those frequencies one wishes to retain. You can also make a band-pass filter by applying a low pass filter to a time series that has already been high-passed (or vice versa), in which case the response function is the product of the two response functions (center case in Figure 8.1). Or you can subtract a low pass filtered version of the data set from another one with a cutoff at a higher frequency, as illustrated on the right of Figure 8.1.

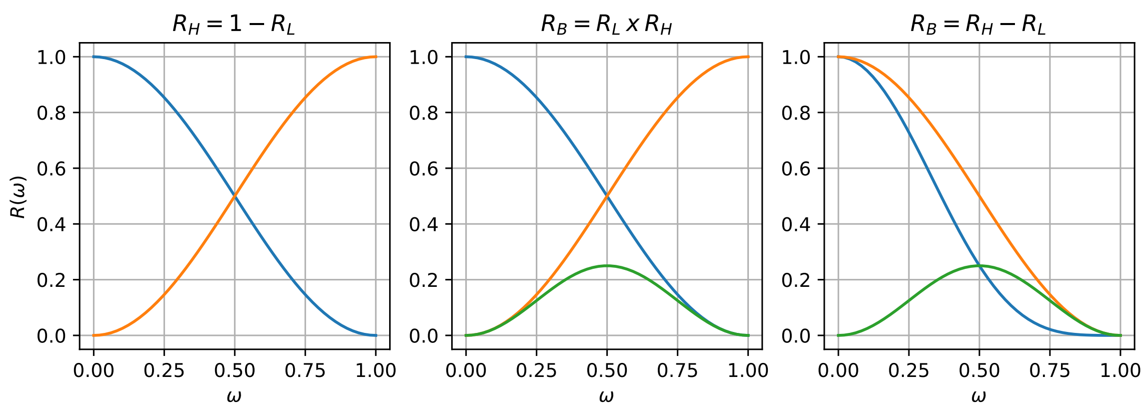

Figure 8.1 Examples of response functions for low(blue), high(orange) and band-pass(green) filters. High-pass can be obtained by subtracting the low-pass from the original data. Band-pass can be obtained by applying both a high-pass and low pass filter, or subtracting low-pass filtered data from low-pass filtered data where the cutoff is at a higher frequency.

8.2.3 Obtaining the Response Function

The response function is the spectrum of amplitude modifications made to all frequencies by the filtering function. It is the ratio of the filtered output amplitude to the unfiltered input amplitude.

$$R(\omega) = \frac{\text{filtered time series at frequency } \omega}{\text{original time series at frequency } \omega} \tag{8.10}$$

Filtering can alter both the amplitude and the phase, so that $R(\omega)$ may be real or imaginary, but we may be interested only in the change of amplitude or power of the response function.

$$|R(\omega)| = \frac{\textit{amplitude of filtered time series at frequency } \omega}{\textit{amplitude original time series at frequency } \omega} \tag{8.11}$$

$$|R(\omega)|^2 = \frac{\textit{power of filtered time series at frequency } \omega}{\textit{power of original time series at frequency } \omega} \tag{8.12}$$

Some filtering is done by nature and instruments and these may introduce phase errors. Filters can be designed with a real response function, unless we desire to introduce a phase shift. Phase shifting filters are not too commonly used in meteorological or oceanographic data analysis or modeling, and so we will not discuss them except in the context of recursive filtering, where a phase shift is often introduced with a single pass of a recursive filter. Centered, symmetric, non-recursive filters have the feature of giving a real response function, and so do not introduce phase changes into the filtered time series.

How do we design a centered, non-recursive set of weights with the desired frequency response? Our smoothing operation can be written,

$$g(t) = \sum_{k=-J}^{J} f(t + k\Delta t)\, w(k\Delta t) \tag{8.13}$$

where $g(t)$ is the smoothed time series, $f(t)$ is the original time series and $w(k\Delta t)$ are the discrete weights applied at 2J+1 time points. In the continuous case we can write this as

$$g(t) = \int_{-\infty}^{\infty} f(\tau)\, w(t - \tau)\, d\tau \tag{8.14}$$

The filtered output $g(t)$ is just the convolution of the unfiltered input series $f(t)$ and the filter weighting function w(t)$w(t)$. From the convolution theorem the Fourier transform of

$$\int_{-\infty}^{\infty} f(\tau)\, w(t - \tau)\, d\tau \quad \text{is} \quad F(\omega)\, W(\omega) \tag{8.15}$$

so that the Fourier transform of the filtered time series is

$$G(\omega) = F(\omega)\, W(\omega) \tag{8.16}$$

So to obtain the Fourier coefficients of the filtered time series we multiply the Fourier transform of the input time series by the Fourier transform of the weighting function. The power spectrum of the filtered time series is thus,

$$P_g(\omega) = G(\omega)\ G(\omega)^* = F(\omega)W(\omega)\ (F(\omega)W(\omega))^* = |F(\omega)|^2\ |W(\omega)|^2 \tag{8.17}$$

From 8.17 we can infer that the response function for the power spectrum is just the power spectrum of the weighting function.

8.2.4 Simple Example of Cosine Wave

Suppose our input time series consists of a single cosine wave with amplitude 1.0. Assuming that the weights are symmetric, the filtered signal is then

$$g(t) = \sum_{k=0}^{J} w(k\Delta t) \cos(\omega(t + k\Delta t)) \tag{8.18}$$

The Fourier Transform of the filtered time series is

$$G(\omega) = 2\int_0^\infty g(t)\cos\omega t\,dt \tag{8.19}$$

Substituting in our expressiom for $g(t)$ we get,

$$\begin{aligned} G(\omega) &= 2\int \sum_k w_k \cos(\omega t + \omega k\Delta t)\cos\omega t\,dt \\ &= \sum_k w_k \cos\omega k\Delta t \\ &= \sum_k w_k \cos\omega t_k \end{aligned} \tag{8.20}$$

For a single cosine wave at frequency ω, the Fourier transform is $F(\omega) = 1$, so that

$$R(\omega) = \frac{G(\omega)}{F(\omega)} = \sum_k w_k \cos\omega t_k = W(\omega) \tag{8.21}$$

The response function $R(\omega)$ is just the Fourier transform of the filter weights. If we assume that the weighting function and the response function are symmetric and real.

$$R(\omega) = W(\omega) = 2\int_0^\infty w(t)\cos\omega\tau\,d\tau \qquad \tau = k\Delta t \tag{8.22}$$

and conversely, the weighting function can be obtained from a specified response function

$$w(\tau) = 2\int_0^\infty R(\omega)\cos\omega\tau\,d\omega \tag{8.23}$$

and $w(\tau)$ and $R(\omega)$ constitute a Fourier transform pair.

8.2.5 The Running Mean Smoother

A popular but very sub-optimal smoother is the running mean smoother, for which the weights are a boxcar function.

$$w(\tau) = \frac{1}{T} \text{ on the interval } 0 < \tau < T \tag{8.24}$$

and zero elsewhere. As we recall from chapter 7, the Fourier transform of a boxcar function is a sinc function.

$$R(\omega) = \frac{\sin\frac{\omega T}{2}}{\frac{\omega T}{2}} \tag{8.25}$$

This response function approaches one at zero frequency, hence it has no effect on frequencies that are very long compared to the averaging interval T. As one can see from Figure 8.2, the response function is zero at every harmonic of the averaging interval T, when $\omega T = 2\pi n, n = 1, 2, 3, ...$ The running mean thus removes exactly all harmonics of the fundamental period T. The running mean has several major weaknesses as a filter, however. It cuts off very slowly and then passes through zero and has a wiggle at higher frequencies that maxes out at about -0.2 near $\omega = 3\pi/2$, and decays very slowly. The sharp cutoff of the weighting function gives rise to Gibbs phenomena in the response function. We might do better to select weighting functions that were more tapered.

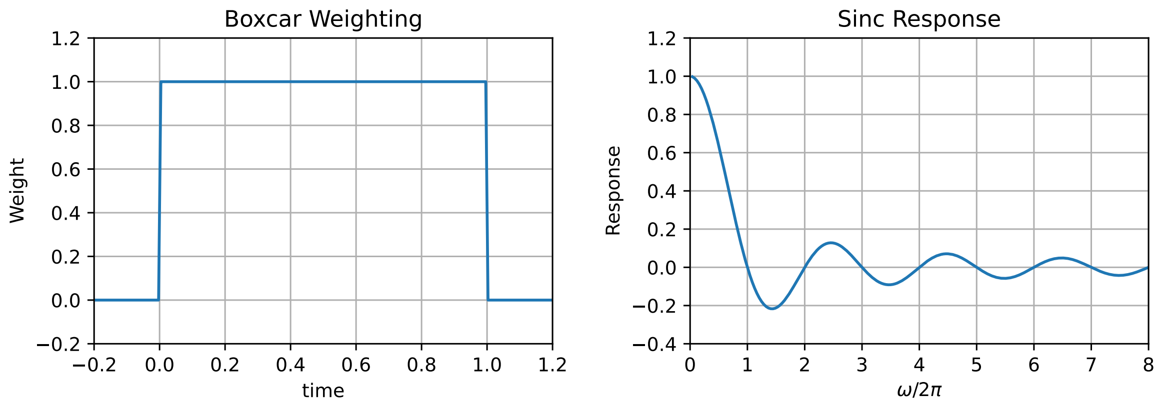

Figure 8.2 Boxcar (running mean) weights and the associated sinc frequency response function.

8.2.6 Construction of Symmetric Non-recursive Filters

In this section we will describe methods for the construction of simple non-recursive filters. Suppose we consider a simple symmetric non-recursive filter.

$$y_n = \sum_{k=-N}^{N} C_k \, x_{n-k} \quad \textit{where } C_{-k} = C_k \tag{8.26}$$

To perform a Fourier transform of (8.26) it is useful to introduce the *Time Shifting Theorem*. Suppose we wish to calculate the Fourier transform of a time series $f(t)$, which has been shifted by a time interval $\Delta t = a$. Begin by replacing t with $t \pm a$ in the Fourier integral representation of $f(t)$,

$$f(t \pm a) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \, e^{i\omega(t \pm a)} d\omega \tag{8.27}$$

which can be slightly arranged to,

$$f(t \pm a) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \, e^{\pm i\omega a} \, e^{i\omega t} d\omega \tag{8.28}$$

from which we infer that the Fourier transform of $f(t \pm a)$ is $F(\omega) \, e^{i\omega a}$. Thus the Fourier transform of the time-shifted time series is the Fourier transform of the original time series multiplied by the factor $z = e^{i\omega \Delta t}$.

We can then Fourier transform (8.26) and use the Time Shifting Theorem to obtain,

$$Y(\omega) = \left[\sum_{k=-N}^{N} C_k \, e^{i\omega k \delta t} \right] X(\omega) \tag{8.29}$$

Here $Y(\omega)$ and $X(\omega)$ are the Fourier transforms of $y(t)$ and $x(t)$, *respectively*. Because the weights are symmetric, $C_{-k} = C_k$, and we know the identity,

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \tag{8.30}$$

we can write (8.29) as,

$$R(\omega) = \frac{Y(\omega)}{X(\omega)} = C_0 + 2\sum_{i=1}^{N} C_k \cos(\omega k \Delta t) \tag{8.31}$$

8.2.7 Frequency Response of Simple Filters

Armed with the simple formula 8.31 we can investigate the frequency response functions of several simple centered, non-recursive filters.

8.2.7.1 The Running Mean Smoother

The running mean smoother replaces the central value on an interval with the average of the values surrounding that point. The running mean can be taken over an arbitrary number of points, e.g. 2, 3, 5, 7. Referring to (8.31) again, a running mean smoother has $C_k = 1/(2N+1$, where $-N < k < N$. The length of the running mean smoother is $2N + 1$. We write below the response functions for running-mean smoothers of length 3,5, and 7. These response functions apply to the Nyquist interval $0 < \omega < \pi/2$

$$\begin{aligned}
2N+1=3: \quad & R(\omega) = \frac{1}{3} + \frac{2}{3}\cos(\omega\Delta t) \\
2N+1=5: \quad & R(\omega) = \frac{1}{5} + \frac{2}{5}\cos(\omega\Delta t) + \frac{2}{5}\cos(2\omega\Delta t) \\
2N+1=7: \quad & R(\omega) = \frac{1}{7} + \frac{2}{7}\cos(\omega\Delta t) + \frac{2}{7}\cos(2\omega\Delta t) + \frac{2}{7}\cos(3\omega\Delta t)
\end{aligned} \tag{8.32}$$

These square weighting functions give damped sine wave response functions (sinc functions), which are generally undesirable, as previously noted. A slightly tapered weighting function, such as the 1-2-1 filter gives a much nicer response function.

$$1-2-1 \ \text{Filter}: \quad R(\omega) = \frac{1}{2} + \frac{1}{2}\cos(\omega\Delta t) \tag{8.33}$$

We have to alter 98.31) a bit to compute the response function for a 1-1 Filter, a running mean that just averages adjacent values. The result is:

$$1-1 \ \text{Filter}: \quad R(\omega) = \frac{1}{2}\cos(\frac{1}{2}\omega\Delta t) \tag{8.34}$$

All these response functions are plotted in Figure 8.3. Note how the 1-2-1 filter cuts off more sharply than the 1-1 filter (running mean 2), but does not have the ugly negative side lobes of the running mean filters and exactly removes the highest resolved frequency. The 1-2-1 is a good simple filter and can be applied multiple times if a stronger low-pass filter is desired.

8.3 General Symmetric Non-recursive Filter Weights

We can find the weights that would give a desired response function as follows. Multiply both sides of (8.31) by $\cos(j\omega\Delta t)$ $j = 0,\ 1,\ 2,\ \ldots N$ and then integrate frequency ω over the Nyquist interval $0 - \pi/\Delta t$

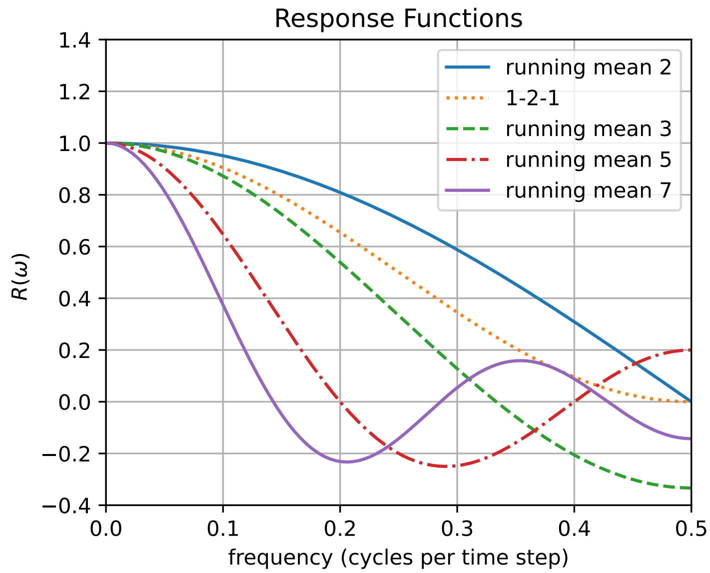

Figure 8.3 Response Functions on the Nyquist interval for various centered, non-recursive filters.

$$\int_0^{\pi/\Delta t} \cos(j\omega\delta t)R(\omega)d\omega = 2C_j \int_0^{\pi/\Delta t} \cos(j\omega\Delta t)\cos(k\omega\Delta t)d\omega \tag{8.35}$$

This yields,

$$C_k = \frac{1}{\pi}\int_0^{\pi} \cos(k\omega')\,R(\omega')d\omega' \tag{8.36}$$

Here $\omega' = \omega\Delta t$, so that $0 < \omega' < \pi$ is the Nyquist interval. From (8.36) we can derive the appropriate weighting coefficient for any arbitrary desired response function $R(\omega)$.

The ideal low-pass filter response function might be one up to some chosen frequency and then cut off abruptly to zero. Let's suppose we want the cutoff to appear at a frequency that is some fraction α $(0 < \alpha < 1)$ of the Nyquist interval, as follows.

$$R(\omega) = \begin{cases} 1 & \omega < \alpha\pi \\ 0 & \omega > \alpha\pi \end{cases} \tag{8.37}$$

Using (8.37) we can perform the integral in (8.36) and obtain,

$$\begin{aligned} C_k &= \frac{1}{\pi}\int_0^{\alpha\pi} \cos(k\omega)\,d\omega \\ C_k &= \frac{1}{k\pi}\sin(\alpha k\omega') \end{aligned} \tag{8.38}$$

Note that the amplitude of the coefficients drops off as k^{-1}, which is rather slow. The coefficients, or weights, C_k, are a sinc function in k, as shown previously in Section 8.2.5. To get a really sharp cutoff of the response function we need to use a large number of weights. Usually we want to keep the number of points to a minimum, because we lose $N-1$ data off each end of the time series and because the computations take time. The computation time problem can be alleviated with the use of recursive filters.

If we truncate (8.37) at some arbitrary value of N, then the response function will be less sharp than we would like and will have wiggles associated with Gibb's phenomenon. This is shown in Figure 8.4, which shows the response functions (8.31) for the weights (8.38) for truncation of N=6, N=12, and N=24. The value of $\alpha = 0.5$ was chosen to cut the Nyquist interval in the center. One can see that more filter weights give a sharper cutoff, but that the Gibb's Phenomena wiggles are undesirable.

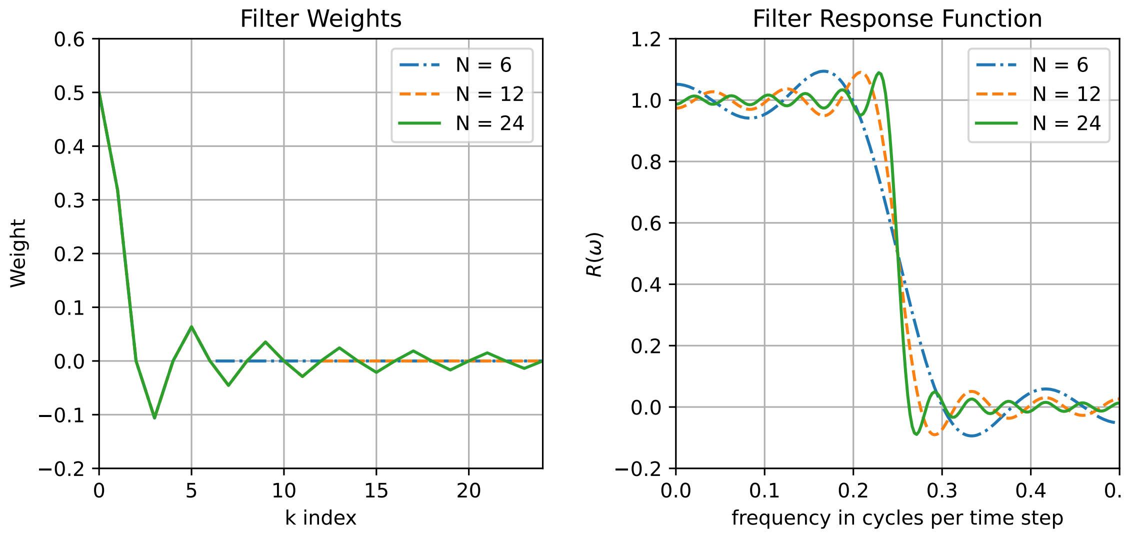

Figure 8.4 Filter weights and response functions for centered non-recursive filters with $\alpha = 0.5$ and N = 6, 12 and 24. Note that only the non-negative k indices are shown.

8.3.1 Lanczos Smoothing of Filter Weights

The wiggles in the response functions of Figure 8.4 have a wavelength of approximately the last included or the first excluded harmonic of (8.38). We can remove this harmonic by smoothing the response function. The running mean smoother exactly removes oscillations with a period equal to that of the length of the running mean smoother. The wavelength of the last harmonic included in (8.38) is $2\pi/N\Delta t$, which suggests that we smooth the response function in the following way.

$$\widetilde{R}(\omega) = \frac{N\Delta t}{2\pi} \int_{-\pi/N\Delta t}^{\pi/N\Delta t} R(\omega) d\omega \tag{8.39}$$

Substituting our equation for the response function (8.4) into (8.39) we get,

$$\begin{aligned} \widetilde{R}(\omega) &= C_0 + \frac{N\Delta t}{2\pi} \int_{-\pi/N\Delta t}^{\pi/N\Delta t} 2 \sum_{k=1}^{N} C_k \cos(k\Delta t\omega) d\omega \\ &= C_0 + \frac{2N}{\pi} \sum_{k=1}^{N} \frac{C_k}{k} \cos(k\Delta t\omega) \sin(\frac{k\pi}{N}) \end{aligned} \tag{8.40}$$

Rearranging this a little, we obtain,

$$\widetilde{R}(\omega) = C_0 + 2\sum_{k=1}^{N} \left\{ \frac{\sin(\frac{k\pi}{N})}{\frac{\pi k}{N}} \right\} C_k \cos(k\Delta t\omega) \tag{8.41}$$

Here we can clearly see that the running mean smoother of the response function has given us a new set of filter weights, where we multiply the original weights by the factor $sinc(\frac{\pi k}{N})$.

$$\widetilde{C}_k = sinc\left\{\frac{\pi k}{N}\right\} C_k \tag{8.42}$$

These factors are sometimes called the sigma factors. Note that the last coefficient, C_N, disappears entirely because the sigma factor is zero ($sin\pi = 0$).

The effect of applying the Lanczos smoothing to the filter weights and response functions are shown in Figure 8.5. The Gibbs phenomenon is greatly reduced and the cutoff is considerably more gradual than for the unsmoothed filter weights. It is clear that one must use a relatively large number of weights to get a good result, since the resulting response function for $N = 6$ is not very functional, although the $N = 12$ $N = 24$ cases look OK.

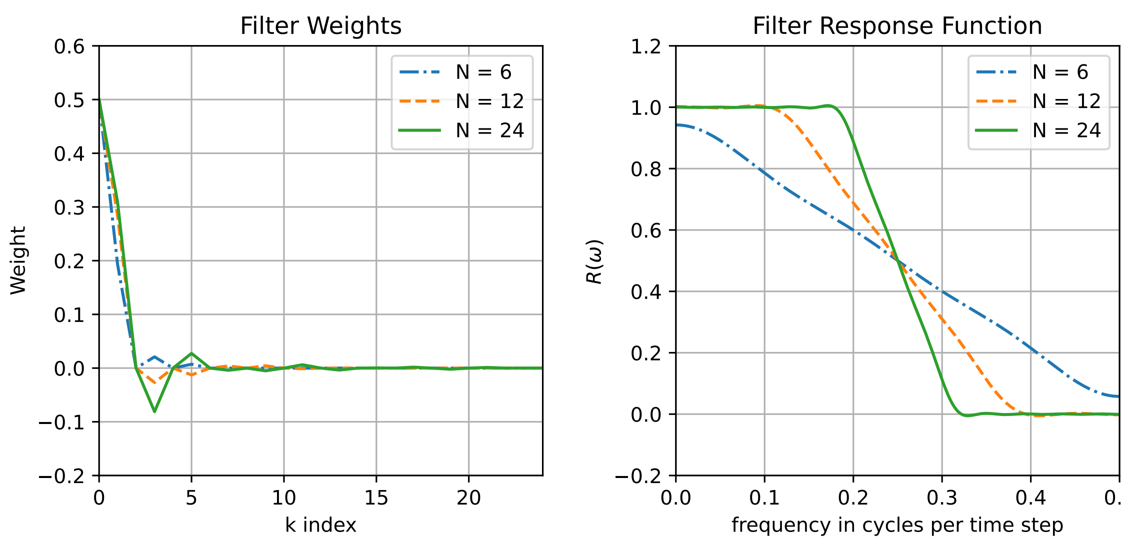

Figure 8.5 Filter weights and response functions for centered non-recursive filters with $\alpha = 0.5$ and $N = 6$, 12 and 24 after applying Lanczos smoothing.

8.4 Recursive Filters

The filters we have discussed so far are obtained by convolving the input series $x(n\Delta t) = x_n$ with a weighting function w_k, in the following way.

$$y_n = \sum_{k=-K}^{K} w_k \, x_{n+k} \tag{8.43}$$

Such filtering schemes will always be stable, but it can require a large number of weights to achieve a desired response function. If greater efficiency of computation is desired, then it may be attractive to consider a recursive filter of the general form,

$$y_n = \sum_{k=-0}^{K} b_k\, x_{n-k} + \sum_{j=-0}^{J} a_j\, y_{n-j} \tag{8.44}$$

In this case the filtered value depends not only on the unfiltered input series, but also on previous values of the filtered time series. In general, sharper response functions can be obtained with fewer weights and thereby fewer computations than with non-recursive filters. The method of constructing the weights for a recursive filter from a desired response function is not as easy as with convolution filters, and the filtering process is not necessarily stable.

8.4.1 Response Functions for General Linear Filters

To find the response function for the general linear filter (8.44) let's first rearrange it in the following way,

$$y_n - \sum_{j=-0}^{J} a_j\, y_{n-j} = \sum_{k=-0}^{K} b_k\, x_{n-k} + \sum_{j=-0}^{J} a_j\, y_{n-j} \tag{8.45}$$

We next take the Fourier transform of (8.45) and use the time shifting theorem.

$$Y(\omega)\left\{1 - \sum_{j=1}^{J} a_j\, z^{-j}\right\} = X(\omega)\left\{\sum_{k=1}^{K} b_k\, z^{-k}\right\} \tag{8.46}$$

Here $z = e^{i\omega\Delta t}$. From this we can obtain the response function.

$$R(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{\left\{\sum_{k=1}^{K} b_k\, z^{-k}\right\}}{\left\{\sum_{j=1}^{J} a_j\, z^{-j}\right\}} \tag{8.47}$$

Here $R(\omega)$ is the system function of the general recursive filter and measures the ratio of the Fourier transform of the output function to the input function. In general $R(\omega)$ will be complex for recursive filters, which means that recursive filters will introduce a phase shift in the frequencies that they modify. This is because the filters are not symmetric, in general. Physically realizable filters, as might be employed to real-time data or in electric circuits, cannot be symmetric, since the future data are not known at the time the filtration of the present value must be produced. In data applications where we want to remove the phase shift, we generally run the filter forward and backward across the data set. The real squared amplitude response function can be obtained from,

$$|R(\omega)|^2 = R(\omega)R(\omega)^* \tag{8.48}$$

where the asterisk indicates the complex conjugate.

8.4.2 A Simple Recursive Filter

We can illustrate some important facts about recursive filters by considering the simple example of a recursive filter given by,

$$y_n = x_n + 0.95\, y_{n-1} \tag{8.49}$$

The response function for this filter can be gotten from the general formula (8.47).

$$R(\omega) = \frac{1.0}{1.0 - 0.95z^{-1}} \tag{8.50}$$

We can find the equivalent non-recursive filter by dividing out the rational factor in (8.50) to obtain a polynomial in z. The result is,

$$R(\omega) = \frac{1.0}{1.0 - 0.95z^{-1}} = 1.0 - 0.95z^{-1} + 0.9025z^{-2} - 0.8574z^{-3} + 0.8145z^{-2} + \dots \tag{8.51}$$

or

$$R(\omega) = \frac{1.0}{1.0 - 0.95z^{-1}} = 1.0 + \sum_{n=1}^{\infty} 0.95^n \, z^{-n} \tag{8.52}$$

Notice how slowly the coefficients of the polynomial decay. These coefficients are also the weights of the equivalent non-recursive filter. Thus many, many points are necessary to replicate the effect of the recursive filter with a non-recursive filter.

8.4.3 Impulse Response of a Recursive Filter

It is important to know how many data points a recursive filter must pass over before its response begins to settle out. This will indicate how many points must be disregarded off the end of a time series that has been filtered recursively. We can address this question by asking how the filter responds to a unit impulse time series of the form.

$$x_n = \begin{cases} 1.0 & n = 1 \\ 0.0 & n \neq 1 \end{cases} \tag{8.53}$$

The time series that results from filtering the time series (8.53) can be called the *impulse response* of the filter. The filter (8.49) acting on the input time series (8.53) will produce the following filtered time series.

$$y_0 = 1.0,\ y_1 = 0.95,\ y_2 = 0.9025,\ y_3 = 0.8574,\ y_4 = 0.8145,\ \dots \tag{8.54}$$

The impulse response(8.54) of the recursive filter (8.49) is identical to the equivalent non-recursive filter and decays very slowly. So we conclude that we lose about the same number of endpoints with both types of filter. The only apparent advantage of the recursive filter is that it requires far fewer computations to achieve the same effect. The phase errors introduced by recursive filters can be reduced or eliminated by passing over the time series twice, once in the forward direction and once backward in time. The resulting amplitude response function is the square of the response function for a single application. Most high-level programming languages (e.g. Python, Matlab) have a filter function that passes twice over the data in this way.

8.4.4 Construction of Recursive Filters

The construction of appropriate weights from a system function, or response function of desired shape is not quite as straightforward for recursive filters as for non-recursive filters, and requires different mathematics. From what we have done so far, it might be obvious that what we need to do is construct the appropriate polynomial in $z = e^{i\omega\Delta t}$ that produces the desired system function as described in (8.47) Recall that z maps into the unit circle in the complex plane, as the frequency varies from zero to the Nyquist frequency. For a recursive filter to be stable, all zeros of the polynomial in the system function must be within the unit circle. It is also useful to realize that the z transform is linear, so that the system function of the sum of two filters is the sum of the system functions for each filter. Also, if two filters are applied successively, the system function of the result is the product of the system function for the two filters.

The construction of recursive filters, that is finding the appropriate weights, can be simplified by transforming the z variable to a w variable defined in the following way.

$$z = e^{i\omega\Delta t} = \frac{1 + iw}{1 - iw} \tag{8.55}$$

or

$$w = i\left(\frac{1 - z}{1 + z}\right) \tag{8.56}$$

8.4.5 Butterworth Filters

As a useful and common example we can consider the Butterworth family of filters with response functions defined as follows.

$$R(\omega)R(\omega)^* = \frac{1}{1 + \left(\frac{w}{w_c}\right)^{2N}} \tag{8.57}$$

The filter has the desirable property of smoothness and high tangency at the origin and infinity. It contains two design parameters, w_c and N, which can be used to design a filter with a cutoff at the desired frequency and the appropriate amount of sharpness to the cutoff. We will not delve further into the specifics of how to derive the filter weights, as most people will simply use an off-the-shelf routine to do this. Butterworth filters are smooth and monotonic, which are generally good characteristics for data work. If a sharper cutoff is required and negative side lobes are tolerable, there are other filtering schemes with these characteristics.

In applying a Butterworth filter, one is given the choice of a cutoff frequency, our parameter α, and an order N. Let us explore the response functions and impulse response functions for several choices of N. Figure 8.6 illustrates how higher orders give sharper response functions, but also very elongated impulse response functions. These features are emphasized as the cutoff is moved to lower frequencies. For $N = 4$ the impulse response function becomes very small after about 20 time steps, while $N = 9$ requires many more time steps for its full effect to be felt.

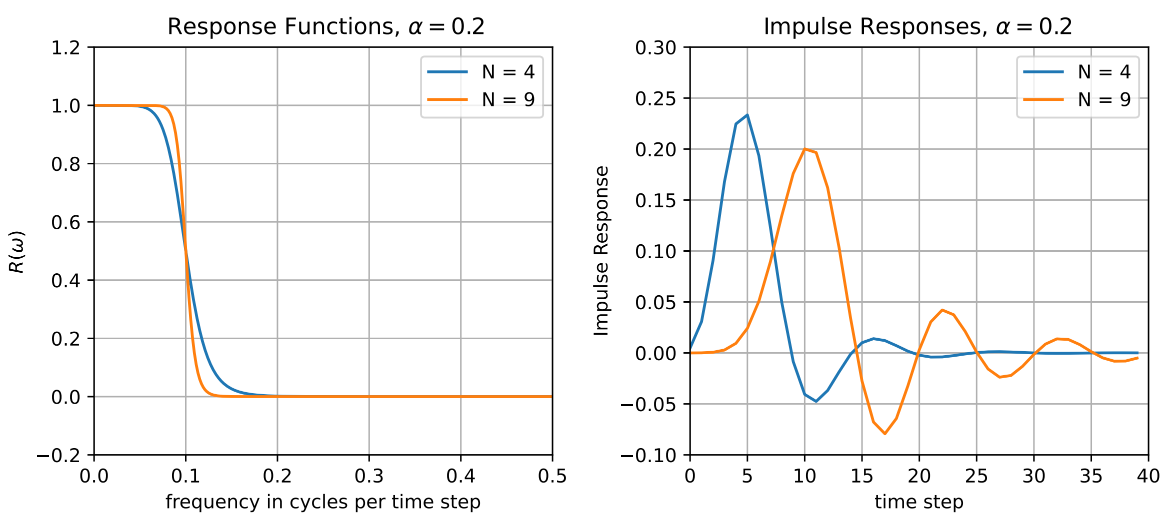

Figure 8.6 Response functions and impulse response functions for Butterworth filters with orders $N = 4$ and $N = 9$, each with $\alpha = 0.2$.

8.4.6 Example using Butterworth Filter

Suppose we have a time series that consists of low-frequency red noise $\mathrm{RN}(t)$ plus a cosine wave with a period of 5 days whose amplitude is inversely proportional to the departure of the red noise from zero. This might be caused by a wave instability that only grows when the mean value of the variable is near zero.

$$f(t) = \mathrm{RN}(t) + \frac{1}{\mathrm{RN}(t) + \epsilon} \cos\left(\frac{2\pi t}{5}\right) \tag{8.58}$$

Here ϵ is a suitably small number. Power spectral analysis of this time series shows red noise with a peak in variance at 5 days, but since the phase information is lost in spectral analysis, we would not learn about the sporadic nature of the 5-day oscillation and its relation to the low-frequency variability (Figure 8.7).

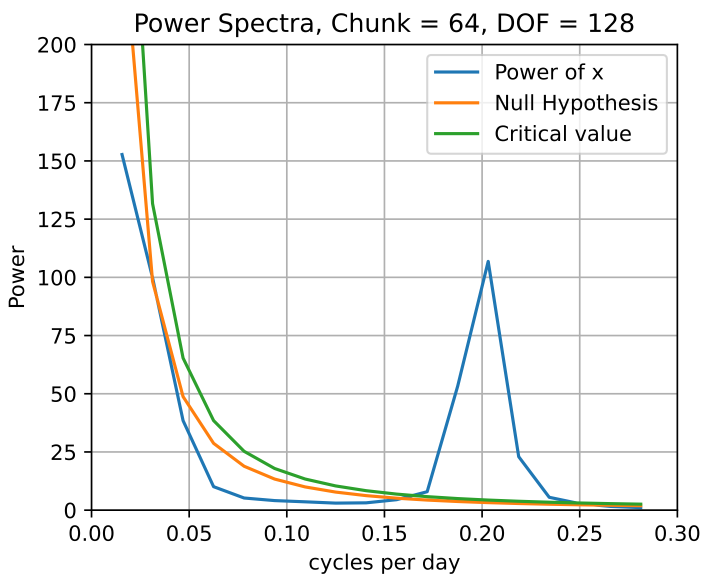

Figure 8.7 Power spectrum of a timeseries with red noise, plus a large and significnt 5-day oscillation.

The power spectrum in Figure 8.7 suggests using a Butterworth filter to separate highpass and lowpass variance at a frequency of about 0.15 cycles per day or $\alpha = 0.3$. We then plot a part of the data showing the lowpass and highpass data in Figure 8.8. One clearly gets the notion that the 5-day oscillation is episodic and has the most amplitude when the low-pass time series is near zero.

To show this another way, in Figure 8.9 we make a scatter plot of the absolute value of the high-passed timeseries versus the low-passed data. This clearly shows a peak in the amplitude of the 5-day oscillation when the low-passed time series is near zero.

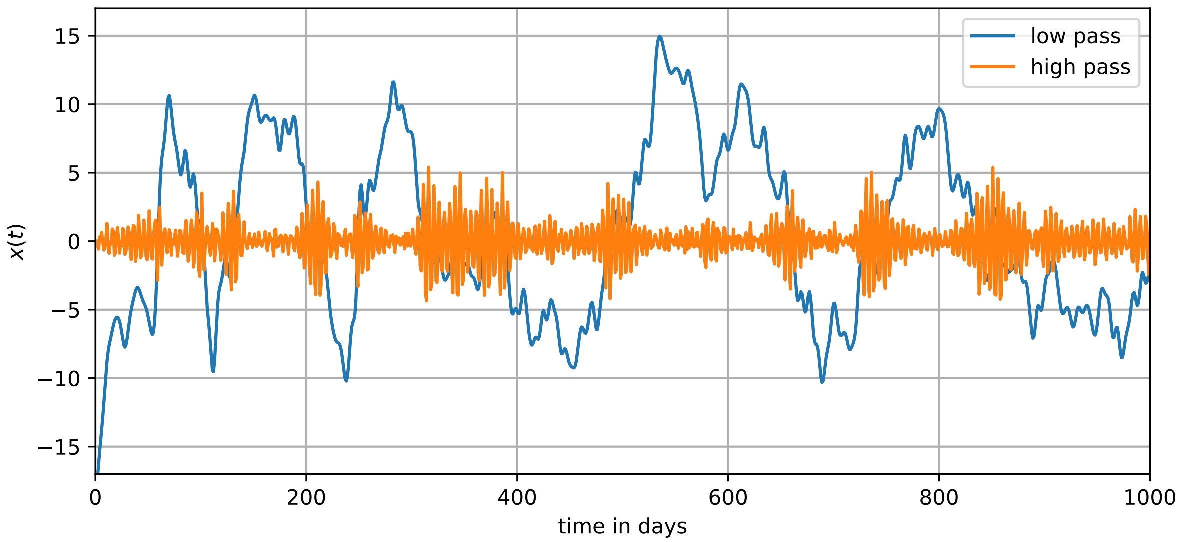

Figure 8.8 The first 1000 days of the high-passed and low-passed timeseries consisting of red noise, plus a 5-day oscillation.

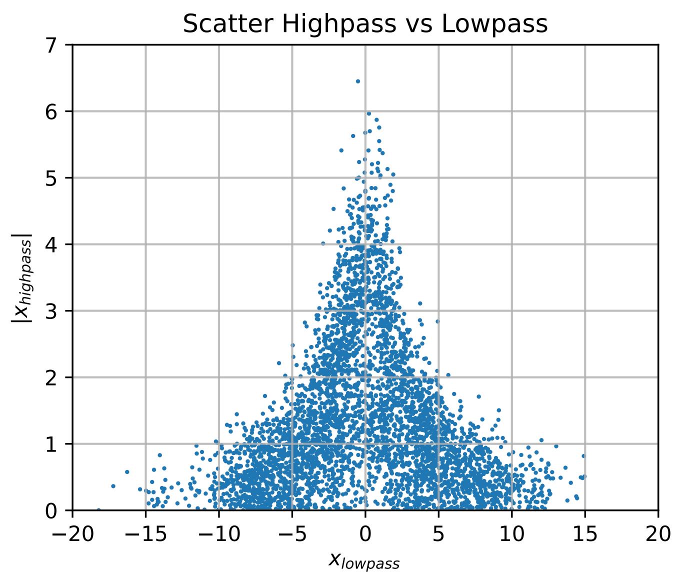

Figure 8.9 Scatter diagram of the absolute value of the high-passed time series versus the value of the low-passed time series.